

11-13-92

204280

NASW-4435

p. 298

Iowa Satellite Project ISAT-1

**Iowa State University of Science and Technology
Iowa Space Grant Consortium**

Summer 1993

Report

July 25, 1993

(NASA-CR-195517) IOWA SATELLITE
PROJECT ISAT-1 (Iowa State Univ.
of Science and Technology) 298 p

N94-24838

Unclass

G3/18 0204280

Iowa Satellite Project ISAT-1

Iowa State University of Science and Technology
Iowa Space Grant Consortium

Summer 1993 Report July 25, 1993

Project Coordinators

William J. Byrd
Dr Leverne K. Seversike

Assistant Project Coordinators

Todd Kuper
Michael Lephart

Integration

Alaina Hathaway
Todd Blummer
Jim Garber

Attitude

Lori Zeimet
Chris Trotta
Michelle Manson

Ground Systems

Mark McKeown
Kelly Parham

Structures

Tad Calkins
Darby Cooper
Sean Olin

Power

Steve Heistand
Ken Longacre

Thermal

Brent Reinders

Payload

Manuel Suarez
Mitchell Suarez
Michael Novy
Chris Silberhorn

Table of Contents

1.	Integration	2
1.1	Introduction	2
1.2	Launch Vehicle	2
1.3	Mass Budget	3
1.4	Power Budget	4
1.5	Packing	5
1.6	ISAT-1 Model	6
1.7	Future	7
1.8	References	7
2.	Structures	9
2.1	Introduction	9
2.2	Structures Performance Criteria	9
2.3	Structures Concepts	12
2.4	Structures Selection Criteria	13
2.5	Preliminary Structures Sizing Analysis	14
2.6	Preliminary Finite element Analysis	14
2.7	Structural Elements	16
2.8	Recommendations for Future Work	25
2.9	References	26
3.	Payload	28
3.1	Earth Radio Frequency Experiment	28
3.2	Ground Transponders	32
3.3	Primary CCD Camera	40
3.4	GPS Unit	43
3.5	Satellite Laser Ranging	44
3.6	Micro-Meteor Detection	45
3.7	Life Sciences	48
4.	Power Systems	53
4.1	Introduction	53
4.2	Solar Array Design	53
4.3	Power Regulation and Control	55
4.4	Batteries	56
4.5	Final Configuration	56
4.6	Future Work	58
4.7	References	59

5.	Thermal Considerations	61
5.1	Introduction	61
5.2	Background	62
5.3	Current Work	62
5.4	Future Work	64
5.5	References	65
6.	Attitude and Control	67
6.1	Introduction	67
6.2	Overview for ISAT-1	67
6.3	Gravity-Gradient Boom	68
6.4	Magnetic Torquers	69
6.5	Sensors	69
6.6	Despin Mode	69
6.7	Capture	71
6.8	Future	71
7.	Ground Systems and Operations	73
7.1	Introduction	73
7.2	Time Line	73
7.3	Data Management	88
7.4	Personnel	92
7.5	Ground Systems Selection	93
7.6	Future Work	94
7.7	References	95
8.	Communication Systems	97
8.1	Summary / Overview	97
8.2	COMM-1	98
8.3	COMM-2	102

Appendix A Figures and Tables

- A.1 Integration
- A.2 Structures
- A.3 Payload
- A.7 Ground Systems and Operations

Appendix B Sources Codes

- B.2 Structures
- B.6 Attitude and Control

Appendix C References Papers

- C.2 Structures
- C.3 Payload
- C.4 Power Systems
- C.5 Thermal Considerations

Table of Figures

2.1	ISAT Internal Configuration	10
2.2	Launch Adapter	11
2.3	Launch Adapter and Panel Overlay	11
2.4	Internal Structure Concepts	12
2.5	ANSYS Internal Element Plot	15
2.6	ANSYS External Element Plot-Panels	15
2.7	ANSYS External Element Plot-Stringers	16
2.8	Joint Detail A	17
2.9	Joint Detail B	18
2.10	Boom Sizing Contour Plot	19
2.11	ANSYS Model	22
2.12	Delta II SPE 1 Orientation	24
3.1	ERFE Block Diagram	28
4.1	Solar Panel Configuration	54
4.2	Main Solar Array	54
4.3	Bracket Solar Array	55
4.4	Solar Array Power	58
4.5	Improved Solar Array Power	59
5.1	Thermal Model	64
6.1	Control Block Diagram	68
6.2	Despin Control	70
8.1	COMM-1 Block Diagram	97
8.2	COMM-2 Block Diagram	97

Table of Tables

1.1	Mass Budget	3
1.2	Power Budget	4
2.1	Static Shelf Load Estimates	14
2.2	Boom Sections Sizes	20
3.1	Weather Station Cost Estimate	39
3.2	Technical Specifications for Canon Ci-20	41
3.3	Kistler 8044 High Impedance Accelerometer	46
3.4	Kistler 5037A Multi-Channel Charge Amplifier	46
4.1	Power Budget	57
5.1	Space Environment	62
7.1	Ground Equipment	93

Introduction

Satellite systems to date have been mainly scientific in nature. Only a few systems have been of direct use to the public such as for telephone or television transmission. Space enterprises have remained a mystery to the general public and beyond the reach of the small business community. The result is a less than supportive public when it comes to space activities.

The purpose of the ISAT-1 program is to develop a small and relatively inexpensive satellite that will serve the State of Iowa, primarily for educational purposes. It will provide products, services, and activities that will be educational, practical, and useful for a large number of people. The emphasis is on public awareness, "space literacy", and routine practical applications rather than high technology.

The initial conceptual design phase was complete when the current team took over the project. Some areas of the conceptual design were taken a little farther, but for the most part this team started at the detailed design stage. This team was split into eight groups:

- Integration
- Structures
- Payload
- Power
- Thermal
- Attitude Determination and Control
- Ground System and Operations
- Communications

Each group's work is expanded upon in the following report. Each chapter relates the group work to date and any considerations that the group felt were necessary for future work.

Chapter 1

Integration

Chapter 1 Integration

1.1 Introduction

The integration team's responsibility was to combine all incoming designs from other teams and combine them into one feasible design. Determine packing considerations in the satellite and construct at least one full scale model of the satellite, for packing considerations. Determine a mass and power budget for the satellite and select a launch vehicle for the satellite.

1.2 Launch Vehicle

The Delta II was selected as the launch vehicle. Several options were investigated including the Space Shuttle, Ariane and Atlas. The Shuttle was not chosen because it doesn't reach a high enough altitude. The Atlas was ruled out because it has no designated secondary payload accommodations and the primary payload option was prohibitively expensive. The Ariane was not used because the launch apparatus and configuration were not compatible with the proposed ISAT-1 design. The only drawback to using the Delta II was that the largest secondary payload envelope offered was not large enough to accommodate a satellite the size of the initially proposed ISAT-1. The size of ISAT-1 was therefore reduced to fit into the available envelope. The current launch date is set for December 1995.

1.3 Mass Budget

The current mass budget is reflected in Table 1.1.

#	Component	Mass (kg)
1	Structure	3.6
1	G. G. Boom	7
2	Communications	3
2	Flight Computer	3
2	Large Experiment Module	3
2	Medium Experiment Module	1.1
6	Small Experiment Module	0.9
1	Pitch Torqrod	0.1
1	Roll Torqrod	0.1
1	Yaw Torqrod	0.45
2	G. G. Boom Tanks	0.1
1	Earth Radio Frequency Experiment	5.4
2	CCD Camera	0.39
1	Telescope	3
6	Battery	0.6
6	Battery Mount	0
2	Magnetometer	0.22

Table 1.1 Mass Budget

1.4 Power Budget

The current power budget is reflected in Table 1.2

#	Component	Power (W)
1	Structure	0
1	G. G. Boom	0
2	Communications	7
2	Flight Computer	7
2	Large Experiment Module	4
2	Medium Experiment Module	1.5
6	Small Experiment Module	1.1
1	Pitch Torqrod	0.25
1	Roll Torqrod	0.25
1	Yaw Torqrod	2
2	G. G. Boom Tanks	0
1	Earth Radio Frequency Experiment	7
2	CCD Camera	5
1	Telescope	0
6	Battery	0
6	Battery Mount	0
2	Magnetometer	0.04

Table 1.2 Power Budget

1.5 Packaging

For the integration and packaging of all the components of the bus subsystems and payloads, the major factor in choosing the components was their size. Also playing a major part in the choice of components was cost. The following is a summary of the bus subsystems to be included in the satellite:

- a. (2) magnetometers, by ITHACO, chosen because of their size. (Figure A.1.1)
- b. (3) Torqrods, by ITHACO, chosen because this was the only place they could be found. There are two 1 a•m² torqrods, and one 10 a•m² torqrod. (Figures A.1.2 and A.1.3)
- c. (6) batteries, by Eagle Picher, which are nickel hydride inside, chosen because of durability. (Figure A.1.4)
- d. (2) flight computers, which will be made by Iowa State University. (Figure A.1.5)
- e. (2) communication units, made by Cynetics. They were chosen because of their relatively small cost. (Figure A.1.6)
- f. (1) telescope made by Iowa Scientific Optical, chosen because they could build it to specifications. The CCD camera to go with it is built by Canon. (Figure A.1.7)

The payloads, up to this point are specified below:

- a. Earth Radio Frequency Experiment (ERFE). (Figure A.1.8)
- b. Two large experiment modules, one which will hold the seed experiment. The other is uncommitted. (Figure A.1.9)
- c. Two medium experiment modules, both which are uncommitted. (Figure A.1.10)
- d. Six small experiment modules, all of which are uncommitted except one, which will hold the electronics needed for the robot arm. (Figure A.1.11)

The experiment modules will be constructed of 6061-T6 aluminum. The satellite structure will be graphite epoxy. The solar cells are gallium arsenide germanium. The packaging arrangement for the satellite is seen in Figure A.1.12. There are three shelves to which the various components can be attached. The satellite is packed in this configuration because it's the only way everything would fit.

1.6 ISAT-1 Model

Part of the integration team's job was to build a full scale mock up of the proposed satellite. This project was started just after mid-term spring semester 1993. As of this date 95% of the payload and 90% of the bus components are complete, just needing to be painted and mounted on the appropriate shelves.

During the planning stages of the model construction, size of all internal components needed to be considered along with the size of the launch vehicle's cargo area. The cargo area defined the maximum dimensions of the satellite and the size of the internal components dictated where they would be placed, besides restrictions due to the component itself, such as the telescope needing to be earthward looking, or the magnetometers needing to be mutually perpendicular. The packing arrangement finally arrived as in Figures A.1.12 and A.1.13. This helped during the finalization of the packing arrangement.

The model is being constructed out of extruded polystyrene and modelers plywood. The polystyrene is being used for the solid models of the bus and payload components, along with portions of the satellite outer surface. Modeler's plywood is being used where load considerations would make the use of polystyrene questionable; such as the shelves, mountings for the bus and payload components, and outer skins. Due to the difficulty in manufacturing cylindrical parts out of polystyrene, the bodies of the batteries were cut from plumbing pipe of an outer diameter matching the actual batteries (2 1/2 inches).

Model construction is expected to be complete by the end of June 1993. This will include two full scale mock-ups, complete with internal bus and payload, and full scale models of all internal components for individual display.

1.7 Future work

Further work for the satellite configuration will happen after every design iteration. The integration team will continue to repackage until the final design. The integration team will also have to repackage probably to account for wiring.

1.8 References

Delta II Launch Planners Manual, Secondary Payload.

Chapter 2

Structures

Chapter 2 Structures

2.1 Introduction

The structure of the spacecraft provides the necessary physical support for all subsystems. The structure must not only contain the payload and bus systems while on-orbit, but must withstand launch conditions and ensure that all of the payload and bus systems will arrive on-orbit in an operational condition.

Previous work on the ISAT structure included the determination of the shape of the exterior. This was selected as hexagonal based on solar cell power calculations. In addition, the maximum dimensions of the exterior were determined based on the launch vehicle envelope chosen (Delta II SPE1). A previous analysis of a metal truss structure had been conducted by the Aerospace Engineering 464 class of Spring 1992. In addition, the internal configuration was provided by the Integration Group.

2.2 Structure Performance Criteria

Several criteria for the performance of the ISAT structure were developed. These included the requirement of supporting the internal payload arrangement as specified by the Integration Group, supporting the McDonnell-Douglas specified SPE1 launch adapter, supporting attitude control mechanisms, and using as little internal volume as possible.

The internal configuration of ISAT-1 is shown in Figure 2.1. This arrangement shows the payload and bus systems arranged on three internal shelves. The payload consists of a series of differing size boxes, for which the mass has been estimated.

The location and arrangement of attitude control mechanisms are also shown in Figure 2.1. Most notably, the structure must be capable of transmitting control torques from the three torque-rods and from the gravity-gradient boom attached at shelf 2.

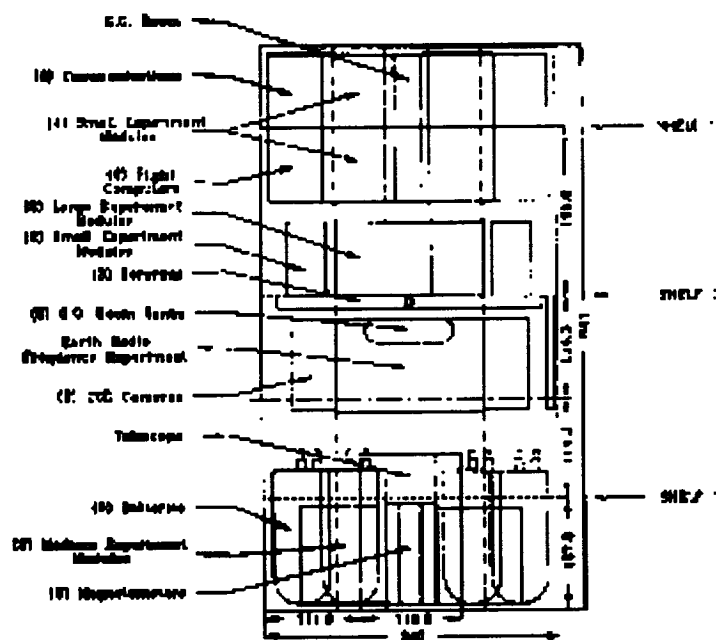


Figure 2.1 ISAT Internal Configuration

The McDonnell-Douglas launch adapter for SPE1 payloads is shown in Figure 2.2. This adapter attaches to the payload with six (6) bolts as shown. This attachment configuration must be supported by one side panel. Figure 2.3 shows the geometry of the launch adapter attachment points with respect to the ISAT panel design.

ORIGINAL PAGE IS
OF POOR QUALITY

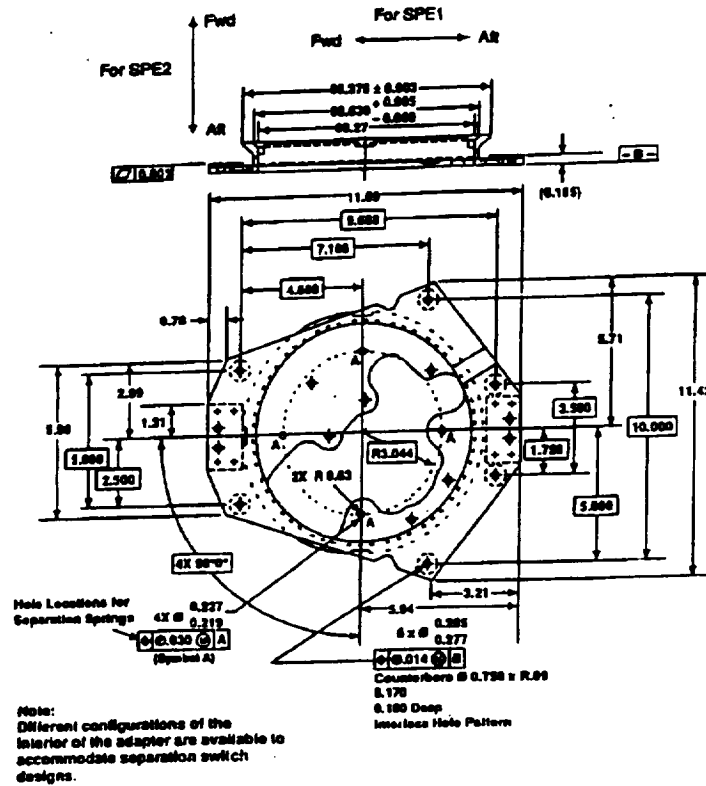


Figure 2.2 Launch Adapter

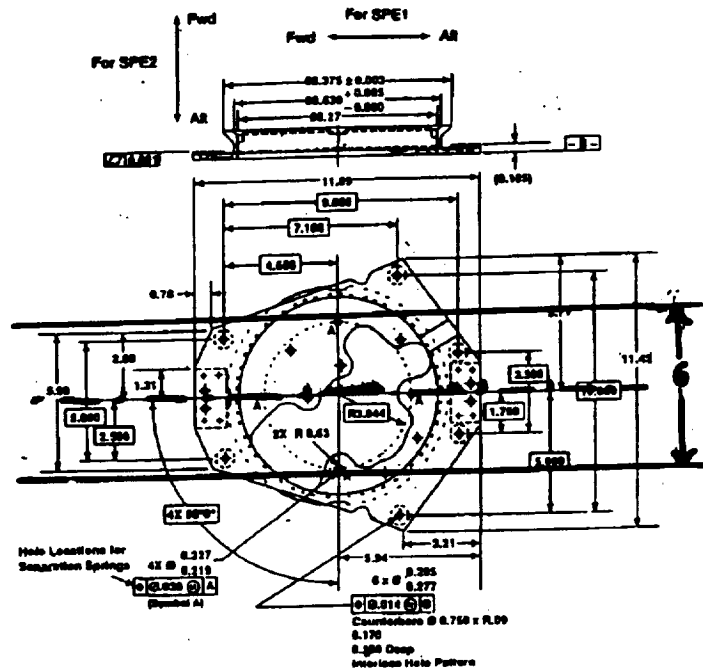


Figure 2.3 Launch Adapter and Panel Overlay

ORIGINAL PAGE IS
OF POOR QUALITY

2.3 Structure Concepts

Five concepts for the internal structure of ISAT were considered. They included Central Support, Modified Central Support, Full Monocoque, Semi-Monocoque, and Truss. These five concepts are shown in Figure 2.4. These structural concepts were considered prior to determination of the internal payload arrangement.

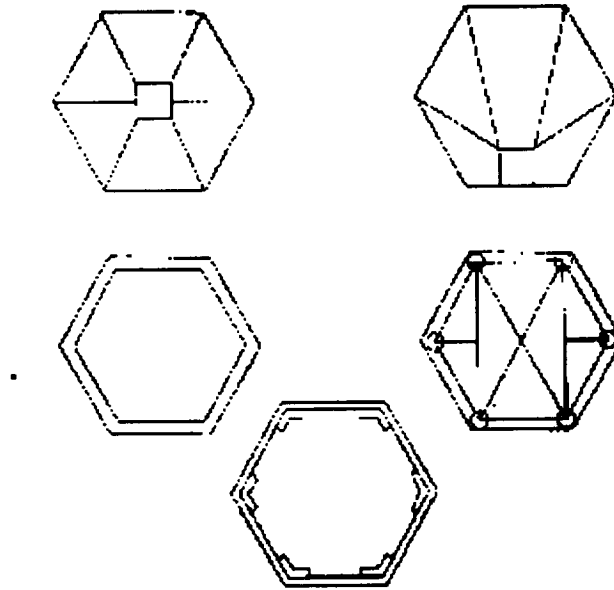


Figure 2.4 Internal Structure Concepts

The Central Support concept would place a large structural member along the central, longitudinal axis of the satellite. Payloads would then be attached to this structural member, with the skin panels being supported by additional braces.

The Modified Central Support concept would place the large structural member along one side of the hexagon structure to accommodate the launch adapter attachment for the Delta II SPE1. Again the payloads would be attached to the major structural member and the skin panels supported by additional braces.

The Full Monocoque concept would consist of thick skin panels that would be designed to support the payloads either by direct attachment or by attachment to cross shelves. The panels would have to be designed thick enough to transmit the launch adapter loadings as well.

The Semi-Monocoque concept would consist of skin panels attached to longitudinal stringers. The design would reduce the panel thickness while transferring more structural material into the corners of the hexagon. Again, one panel would have to be designed to accommodate launch loads.

The Truss concept would consist of thin skin panels attached to a framework of structural elements. This framework would support the internal arrangement of payloads.

2.4 Structure Selection Criteria

One of the major considerations in the final selection of a structural concept was the amount of internal volume that would be required. It was felt that the semi-monocoque structure would best fulfill these criteria, as it would make the best utilization of previous unused space in the corners of the hexagon shape.

Another consideration was the ability of the structure to be adapted to the unique launch interface. The placement of the launch interface on the side of the satellite, require that non-uniform loading be a consideration. It was felt that the semi-monocoque structure would again adequately meet this consideration. The semi-monocoque structure can be adapted to the launch adapter by varying the stringer and panel thickness on the side of the launch adapter.

Fabrication and assembly of both the structural components and the integration of the payload were also considered. The semi-monocoque concept provides for greater flexibility in fabrication and assembly sequences than the other concepts. It should be possible to design the structure such that all segments of the payload are accessible even with the launch adapter in place.

2.5 Preliminary Structure Sizing Analysis

A preliminary analysis of the structure was conducted for sizing purposes. For this case, each payload was considered to be of uniform mass, and its mass would act at its center of mass on each shelf. These loads were then transferred to the six stringers for each shelf. The details of this analysis are presented below in Table 2.1. The mass budget may be seen in Table A.2.1.

Bottom	0.0
Shelf 1	92.4 Newtons
Shelf 2	138.5 Newtons
Shelf 3	156.0 Newtons
Top	to be determined

Table 2.1 Static Shelf Load Estimates

These loads were then used to conduct a stress evaluation of the stringers and the shear stress in the panels. This analysis was based on the asymmetric beam bending equation.

The program calculated the axial stress in the stringers and the shear stress in the panels due to the maximum shelf loading case. The code may be found in Appendix B.2. Results indicated that for the static case, a panel thickness of 0.040 inches and a stringer with flanges 15% of the panel width and a thickness of 0.080 inches should be adequate.

2.6 Preliminary Finite Element Analysis

The results from the preliminary sizing analysis were used to construct a finite-element model. This model used the panel and stringer thicknesses mentioned above. The element plot for the internal configuration is shown in Figure 2.5. The element plot for the external panel configuration is shown in Figure 2.6, and the figure for the external stringer configuration is shown in Figure 2.7. An additional element plot may be found in Figure A.2.1. The model was

fixed in rotation and translation at the four launch adapter attach points. The model was loaded with the estimated shelf loads.

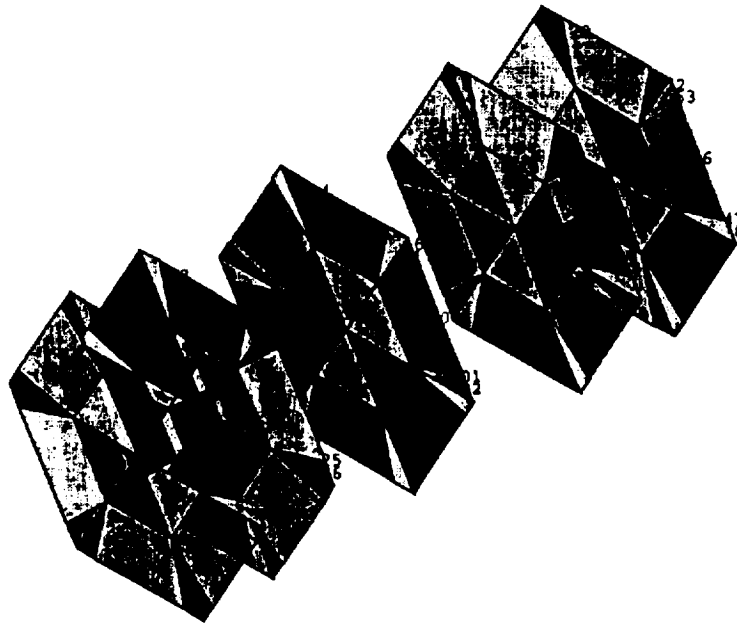


Figure 2.5 ANSYS Internal Element Plot

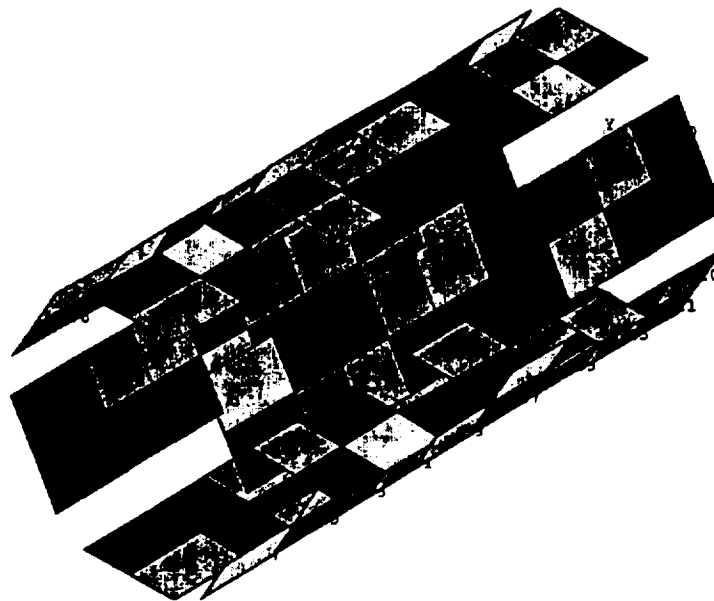


Figure 2.6 ANSYS External Element Plot - Panels

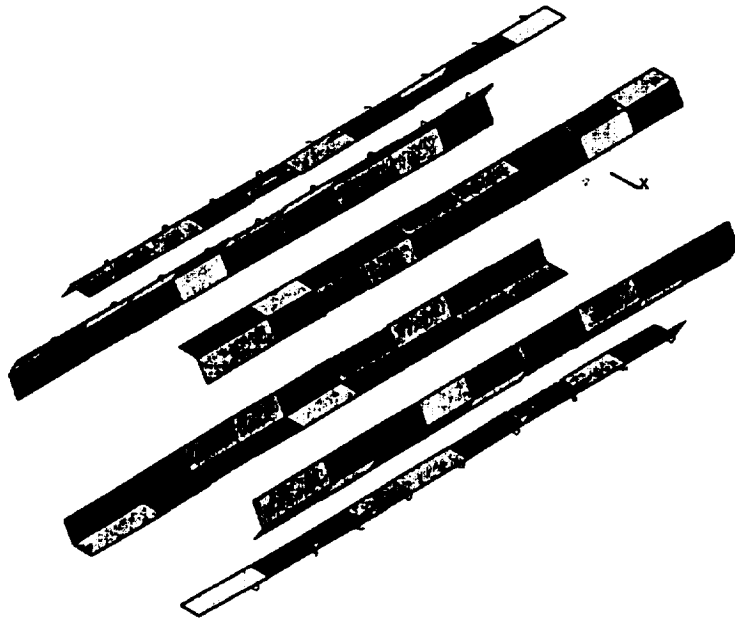


Figure 2.7 ANSYS External Element Plot -- Stringers

2.7 Structural Elements

2.7.1 Gravity-Gradient Boom Design

The structures group also performed the design of the gravity gradient boom. For additional information on the Gravity-Gradient Boom see Appendix C.2.

2.7.1.1 Boom Section Selection

The closed-section design was selected for further development. This section was chosen for its torsional properties and its uniform bending stiffness. The simplicity of the deployment mechanism was also a consideration. The current assumption is that the magnetic portion of the satellite attitude control system will be able to invert the satellite with the boom deployed, so the boom will only be deployed once.

Figure 2.8 and Figure 2.9 show the details of the joint design. Figure 2.8 is a cut-away of the outer wall of a section. Figure 2.9 is a view of an inner section, with the outer section not shown. This design provides for a forward stop collar located on the inner wall of the outer boom section, and a series of forward segmented stops located on the outer wall of the inner boom section. A segmented stop collar is also located on the inner wall of the outer section. A rear stop collar is also located on the inner section. When pressurized, the boom extends until the forward stop collar on the outer section comes into contact with the segmented collar on the inner section. The segments on the inner section slide through the gaps in the collar on the outer section, and are then rotated to lock the boom in its deployed state. The joint design also provides for the placement of three gaskets: one each at the base of the inner section, the top and base of the rear stop collar to provide a gas seal for the pressurized deployment of the boom.

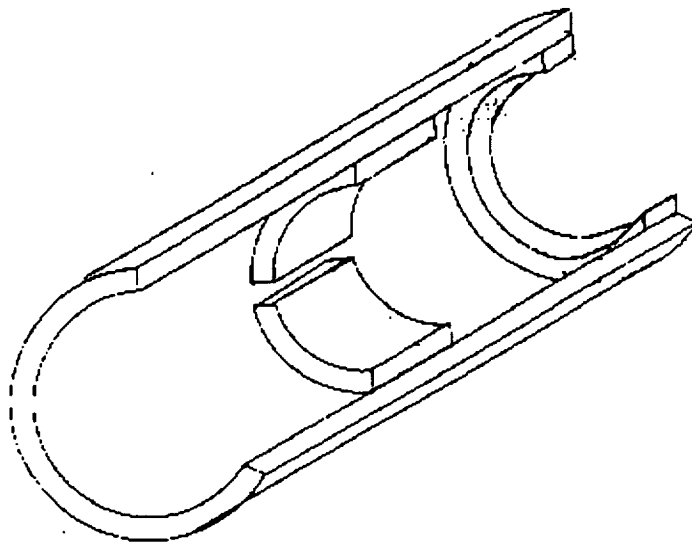


Figure 2.8 Joint Detail A

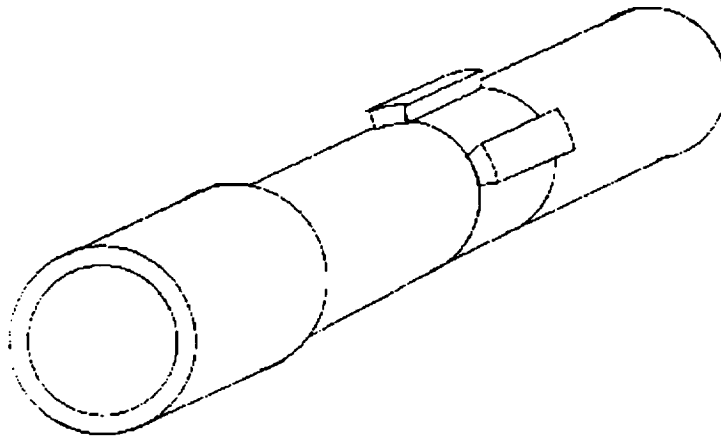


Figure 2.9 Joint Detail B

2.7.1.2 Boom Sizing

The operational task of a gravity-gradient boom is to modify the inertia properties of a spacecraft to increase its stability. The degree of stability increase desired is the driving factor in sizing the boom's length and tip mass to a particular satellite. Iowa State University is currently undertaking an effort to design, construct and launch a small satellite, ISAT-1 (Iowa Satellite One). The preliminary design of this satellite was used to size the boom length and mass for this project.

Current efforts of the Iowa Satellite Project provided much of the needed information about the satellite attitude dynamics. The attitude determination and control group provided the mass moment inertias of the spacecraft body. These inertia calculations assumed uniform mass distribution within the satellite. A design sizing code with boom length and tip mass as control variables was written to determine an optimum length and tip mass for the satellite. The code allows the user to select the desired stability in terms of the stability criterion Θ_x . The user also selects the desired range of boom lengths and tip masses. The code then iterates through these two variables, calculating the new mass moment of inertia I_x , and uses the new inertia to calculate the value of Θ_x . The calculated value of Θ_x is subtracted from the target value, and the absolute value of the difference is

written to a data file along with the boom length and mass. This information can then be plotted on a contour plot. Figure 2.10 shows the contour plot for a stability criterion (Θ_x) of 0.98. Several candidate designs meet the desired stability. The final boom size was selected to be 3 meters in length with a 6 kilogram tip mass. This represents almost a fifteen fold increase in I_x .

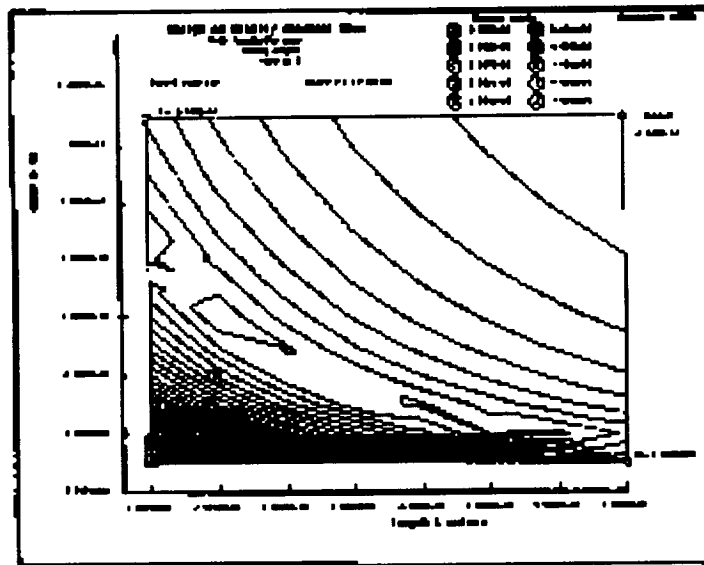


Figure 2.10 Boom Sizing Contour Plot

Stability considerations determined the boom length and tip mass. The diameter and thickness of each section would be determined according to the loads acting upon each section. Loads acting on the gravity-gradient boom were modeled as the Earth's gravitational force acting on the tip mass. This in turn caused forces and moments to be exerted on the boom. A section sizing code was written to size the boom sections based on the known loads, the material properties of graphite-epoxy composites, and a selected factor of safety. It was not possible to accurately determine dynamic loads on the gravity-gradient boom due to insufficient source data from the ISAT project, so the factor of safety used in the sizing analysis was increased.

The code was set up to allow the user to choose the desired tip mass and boom length for section sizing analysis. The forces and moment on the boom were then determined based on the user's input of the tip mass and the maximum

deflection of the satellite from the vertical axis. Once the user had completed the inputs of number of sections, minimum section thickness, gap between sections, and initial base section outer diameter, the code would size the boom sections. This entailed determining the necessary outer diameters, section thicknesses, and inner diameters corresponding to each boom section. If the section thickness needed was found by the code to be smaller than the minimum input section thickness, the code would substitute in the minimum section thickness and continue the analysis. This measure was included in order to allow for more reasonable composite layup thicknesses in the fabrication process.

Results of section sizing yielded necessary section thicknesses for a number of base section outer diameters. A base section having an outer diameter of 0.0381 meters was chosen based on the limited volume fraction of ISAT-1 that would be required. Table 2.2 shows the section sizes in terms of outer diameters, thicknesses, and corresponding inner diameters. It should be noted that in all cases the thicknesses used are the minimum section thickness, and the gap between sections is constant for ease of fabrication.

Boom Section (#)	Outer Diameter (m)	Section Thickness (m)	Inner Diameter (m)
1	0.0381	0.00102	0.0371
2	0.0361	0.00102	0.0351
3	0.0340	0.00102	0.0330
4	0.0320	0.00102	0.0310
5	0.0300	0.00102	0.0290
6	0.0279	0.00102	0.0269

Table 2.2 Boom Section Sizes

2.7.1.3 Fabrication Methods

Fabrication methods were developed to produce the necessary sections to conform to the design. This included the construction of suitable mandrels and

development of layup procedures. This also included developing a fabrication technique for the section joints.

The mandrels used for curing of the boom sections consisted of two sections of steel pipe, each 0.61 meters in length. The two pipes had inside diameters of 0.0343 meters and 0.0394 meters. These pipes were split along their length at the diameter. These were complemented by three rubber hose sections of 0.76 meters in length. The hoses had outside diameters 0.0262 meters, 0.0315 meters, and 0.0361 meters. This provided the capability to fabricate boom sections of varying diameter and thickness. The hoses were fitted internally with steel rods along their entire length. Both ends of the hoses were sealed, one containing a valve, to allow pressurization during the curing process.

Each section was fabricated from three plies of woven $[0^\circ, 90^\circ]$ graphite-epoxy composite. The composite material was cut to a length of 0.56 meters and a width equal to the circumference of the desired section. The three plies were then debulked with the edges staggered prior to being placed in the mandrel. The debulked composite was then placed in the mandrel and wrapped around the hose. Finally, the pipe section was clamped around it. Bleeder cloth was inserted between the composite material and the outer pipe as necessary to form each section.

With the mandrel clamped together, the inner hose was pressurized to 275,790.3 Pa (40 psig), and the mandrel was placed in an oven to cure for 3 1/2 hours at 176° C. Upon completion of the cure, the mandrel was disassembled, and the completed boom section was removed.

To avoid the added complexity and difficulties involved with co-curing the joints along with the respective boom sections, joints from separate cure cycles were epoxied in at a later time. A suitable gas seal was added to each section, and the boom was assembled.

2.7.1.4 Testing

Once the boom was complete, testing was conducted to ensure that the mechanical performance of the gravity-gradient boom design was satisfactory. The first test to be run was an analytical test involving ANSYS finite-element analysis. The boom was modeled on ANSYS using plate elements. A static analysis was performed. Figure 2.11 gives a representation of the deflection resulting in a maximum load condition applied to the boom and its tip mass. The boom was found to have a maximum deflection at the tip of 0.05 meters (1.969 inches).

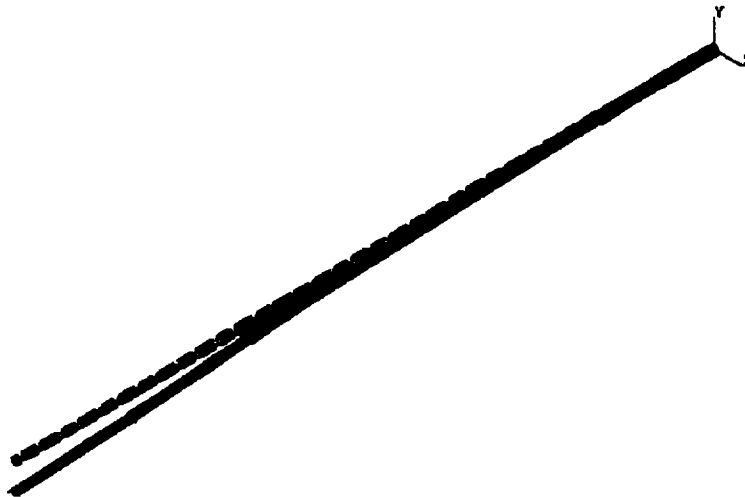


Figure 2.11 ANSYS Model

2.7.2 Shelving

2.7.2.1 Design Considerations

Each shelf must be capable of supporting all loads and payloads. In addition the shelves must accommodate the payload boxes, including any unique interfaces (radiator, tape, electrical/power connections, etc.). Some payloads may

also require vibration isolation. The internal volume of the spacecraft and the payload requirements limit the allowable deflection for each shelf. Other concerns include thermal conductivity, vibration characteristics, cost, weight, ease of fabrication, and ease of access.

2.7.2.2 Design

Three general types of shelving are available: solid aluminum, or graphite epoxy with cut out sections, stringers for supporting payloads with skin as needed, thin honeycombed construction.

2.7.2.3 Assumptions

Preliminary modeling and analysis of the shelves assume that the payload containers carry no load. The payloads are modeled as point masses on appropriate elements of the shelf.

2.7.2.4 Results

The preliminary shelf modeling was included in the ANSYS finite element model, Appendix B.2. Detailed analysis of individual shelf behavior has not been completed. The ISAT structural mass estimate may be seen in Table A.2.2.

2.7.3 Launch Adapter

The launch adapter ring provided by McDonnell-Douglas for the SPE1 payload fairing on its Delta II rocket is shown in Figure 2.2. This adapter attached to the launch vehicle in the payload envelope shown in Figure 2.12. The ISAT structure must be designed to accommodate this launch adapter and fit within the designated payload envelope.

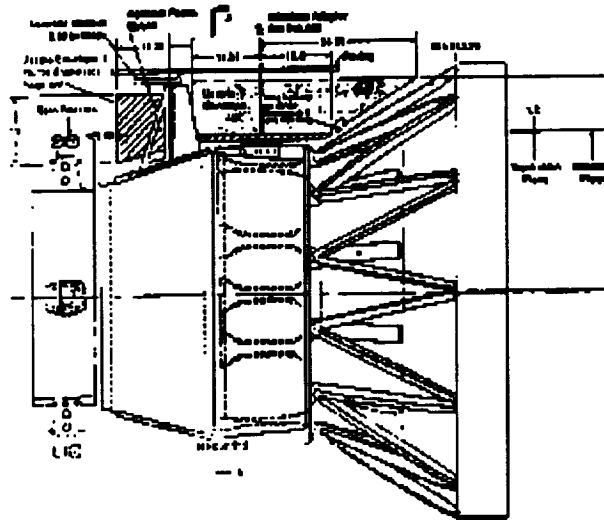


Figure 2.12 Delta II SPE1 Orientation

2.7.3.1 Payload Envelope Considerations

The ISAT main structure has been sized to fit within the allowable payload envelope. This results in a regular hexagon with a side length of 0.170 meters, and a height of 0.643 meters. This size is shown in Figure 2.1. The allowable and negotiable space must also accommodate the gravity-gradient boom that extends above the satellite 0.2134 meters without the tip mass.

2.7.3.2 Launch Adapter Considerations

The launch adapter design is fixed by McDonnell-Douglas Space Systems. The adapter consists of a ring that is clamped to the launch vehicle. Six attachment bolts are provided to attach the launch adapter to the payload. The ISAT structure must be designed to accommodate the given interface and be able to jettison the launch adapter when orbit is achieved. The size of the ISAT panel is fixed by the size of the launch envelope. The width of the launch adapter exceeds that of one side panel. Therefore, two of the attachment bolts will not directly contact any side panel. The geometry of the launch adapter and the ISAT side panel is shown in Figure 2.3.

Several designs have been considered to meet this challenge. One possibility is the reinforcement of the side panel facing the launch adapter by addition of stringer, increasing the size of existing stringers and/or increasing panel thickness. This design may make it possible to not use the two attachment bolts not directly in contact with the panel. Further study of this design should require consultation with McDonnell-Douglas.

2.8 Recommendations for Future Work

The dynamics loads during the launch phase are of primary concern. In addition, on-orbit attitude dynamics such as tumbling, de-spin, and inversion need to be characterized and modeled.

A detailed design of the launch adapter attachment to the satellite needs to be conducted to determine the transfer of loads from the attachment points to the remainder of the structure. McDonnell-Douglas needs to be consulted concerning the possibility of using only four of the six available launch adapter attachment bolts.

A more detailed study of the trade-offs involved with panel thickness and stringer area needs to be conducted.

The boom attachment mechanism and location of support hardware need to be designed.

Additional research is necessary to understand the effects of the space environment on composite materials.

Additional structural requirements, such as deflection limits, thermal considerations, payload interfaces, and payload geometry, need to be compiled from other groups.

Further work is needed to understand the application of graphite-epoxy composites to gravity-gradient booms. The feasibility of the application has been demonstrated. However additional development work and testing are needed to provide the validation necessary to prepare hardware for use in a space application.

2.9 References

Aircraft Structures, David J. Peery and J.J. Azar, McGraw-Hill Book Company, New York, NY.

Roberson, R. E., "Gravitational Torques on a Satellite Vehicle," *J. Franklin Inst.* **265**, 12-33 (1958).

Spacecraft Attitude Dynamics and Control, Vladimir A. Chobotov, Krieger Publishing Company, Malabar, Florida (1991).

Modern Spacecraft Dynamics and Control, Marshal H. Kaplan, John Wiley and Sons, Inc., New York, New York (1976).

Hunt, J.W., and Ray, J.C., "Flexible Booms, Momentum Wheels, and Subtle Gravity-Gradient Instabilities," AIAA Paper No. 92-1673 (1992).

Peterson, M.R., and Grant, D.G., "The Polar BEAR Spacecraft," *Johns Hopkins APL Tech. Dig.* Vol. 8, no. 3, 295-302 (1987).

Composite Materials for Aircraft Structures, Brian C. Hoskin and Alan A. Baker, Ed., American Institute of Aeronautics and Astronautics, Inc., New York, New York (1986).

Space Materials Handbook, John B. Rittenhouse, et al., Lockheed Missiles and Space Company, Palo Alto, CA (1968).

Space Mission Analysis and Design, Wiley J. Larson and James R. Wertz (editors), Kluwer Academic Publishers, Dordrecht, The Netherlands (1992).

Vibration of Mechanical and Structural Systems, M.L. James et al., Harper & Row Publishers, Cambridge, MA (1989).

Formulas for Natural Frequency and Mode Shape. Robert D. Blevins, Robert E. Krieger Publishing Company, Malabar, FL (1979).

Chapter 3

Payload

Chapter 3 Payload

3.1 Earth Radio Frequency Experiment

3.1.1 Introduction

The Earth Radio Frequency (ERF) Experiment will measure the intensity and spectrum of terrestrial communication signal "leakage" through the Earth's ionosphere over the frequency range of 1.5 - 34.5 MHz. This experiment will collect data that, when analyzed on the ground, will help determine the spectrum, intensity, and temporal characteristics of this signal leakage. This data will be used to determine whether low frequency radio astronomy is feasible from Earth orbit.

The ERF Experiment consists of a swept LO heterodyne receiver with analog-to-digital conversion, some basic signal processing, and data storage elements, Figure 3.1.

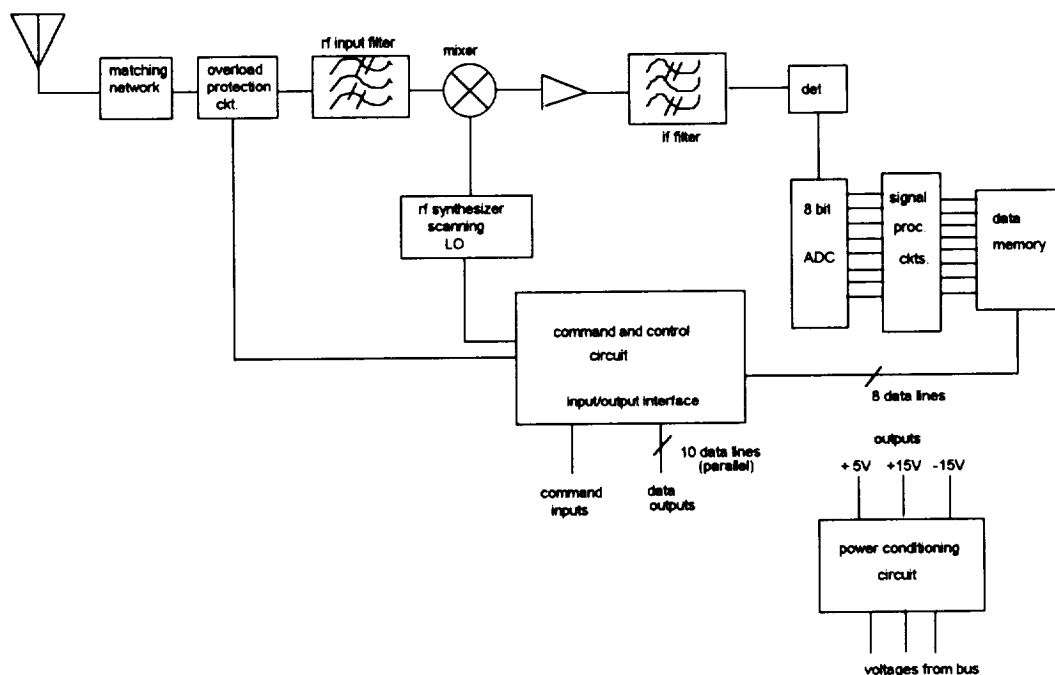


Figure 3.1 ERFE Block Diagram

3.1.2 Mechanical Requirements

The ERF receiver shall be designed and built to withstand the hostile environment or space and in such a manner that it will successfully accomplish its mission. The following paragraphs provide detailed mechanical design requirements to meet these goals.

3.1.2.1 Physical Dimensions

The experiment shall be contained in an enclosure measuring 6" wide by 8" long by 4" high.

3.1.2.2 Enclosure

The enclosure containing the ERF experiment shall be made of 0.25" thick machined aluminum stock to provide adequate radiation shielding for the RF, analog, and digital circuitry inside. The enclosure shall be constructed in two sections. When the receiver is operational, the circuitry must be sealed inside the enclosure using non-outgassing materials.

3.1.2.3 Connectors

3.1.2.3.1 Power

The connector providing the supply voltages to the receiver from the power bus shall be of (TBD) type, with the following pin output:

- 1) +5V
- 2) +15V
- 3) -15V

3.1.2.3.2 Antenna/RF Input

The antenna/RF input connector shall be a 50 Ω , type (TBD) connector.

3.1.2.3.3 Digital Data Input/Output

The connector(s) for the receiver's digital command inputs and data outputs shall be the (TBD) pin D sub miniature type, with built-in filtering.

3.1.3 Environmental Requirements

3.1.3.1 Temperature Range

The receiver must meet performance requirements over the following temperature ranges:

-30° to +70° C	operational
-40° to +80° C	non operational

The receiver must be capable of functioning within design specifications after withstanding temperatures in the non operational range.

3.1.3.2 Vibration

(to be determined)

3.1.3.3 Radiation

(to be determined)

3.1.4 Power Consumption

The receiver shall have a maximum power consumption of 7W while operational. Its standby power consumption shall be (TBD) W.

3.1.4.1 Power Conditioning

The receiver should be capable of conditioning the power received from the spacecraft. The available input voltages shall be :

- | | |
|------------------------|-----------------------------|
| + 5 V \pm (TBD) % | digital circuitry voltage |
| \pm 15V \pm (TBD)% | analog/RF circuitry voltage |

3.1.4.2 Power Stability

The power conditioning circuitry in the receiver shall provide a stable source of power to the instrument. The voltages shall not vary by more than (TBD)% over a period of 12 hours.

3.1.5 Future Work

Additional work is required in the following areas:

- 1) Connector pinout compatibility
- 2) Digital command and output data format
- 3) Proof-of-concept RF circuit

3.2 Ground Transponders

3.2.1 Introduction

The power/transponder will provide a ground based measurement/datalogging capacity to ISAT-1. These transponders will have the capability of communicating with ISAT-1 to upload previously logged data. All remote ground transponders will be based upon the general power/transponder unit.

3.2.2 Power/Transponder Unit

The power/transponder unit will allow a common component to all configurations of the remote ground transponder. It will provide communications capability that is independent of the data to be uploaded to ISAT-1 and will provide the power required for communications. The environment is assumed to be harsh with a possible temperature range of -35°C to 50°C (-31°F to 122°F).

The purpose of the ground transponder will be to uplink data gathered from ground sensors when the satellite comes in the coverage region.

3.2.2.1 Purpose (assuming FDMA)

- a. A receiver installed in the transponder would be constantly checking for a pilot signal (constant frequency tone) from the satellite.
- b. When it receives the signal, the transmitter would be switched ON and data coming from a data storage bank would be uplinked to the satellite.
- c. This would continue until the pilot signal level falls below certain threshold, indicating that the satellite has moved away from the coverage zone.

3.2.2.2 Data Rate / Bandwidth

Average number of passes per day	= 4
Average contact time per pass	= 8 min.
Approximate number of sensors per transponder	= 10
Number of quantization levels	= 1024
Number of data bits per sample	= 10
Time between two samples	= 15 min.
Total data generated per day	= 96 k bits
Total data to be uplinked per pass	= 24k bits
Uplink data rate (assuming FDMA)	= $24k/480 = 50$ bps
Bandwidth requirement per transponder	= 50 Hz
Frequency spacing between two transponders	= 10 Hz
Bandwidth requirement for 200 transponders	= 12 k HZ

3.2.2.3 Frequency Selection

Selection of frequency is based upon the following considerations:

- Antenna size is smaller and thus easily manageable at higher frequencies.
- Free space loss is lesser at lower frequencies.
- Doppler frequency shift is higher at higher frequencies.
- Operating frequency of COMM-1 and COMM-2 should be close enough to share a common antenna for on-board communication equipment.
- Ultimate frequency selection is subject to the availability of a particular frequency band and the FCC location.

3.2.2.4 Reliability / Redundancy

Since this is an experimental communication system and also the type of data is such that certain level of transmission errors can be tolerated, the maximum Bit-Error-Rate is 10^{-5} which is lower than that of COMM-1 which was 10^{-6} .

Also the transponder would remain on ground and hence the equipment redundancy is not critical. However since these transponders would operate as unmanned up-link stations under normal weather conditions, they are expected to have a mean-time between-failure of at least few months.

3.2.2.5 Weight / Size

Again since this is not an on-board equipment, the weight/size considerations are not very critical. The transponder size is expected to be around 10 lbs while the antenna weight/size is largely dependent upon frequency of operation with a lower limit of 10 lbs and an upper limit of 50 lbs.

3.2.2.6 Antenna Design

In order to minimize system complexity a fixed antenna with relatively wide beam width is considered suitable. The antenna polarization would be circular.

Antenna type: Two Yagi Arrays arranged perpendicular to each other.

Antenna Elements:	Central supporting boom	1
	Reflectors	1 for each array
	Dipoles	1 for each array
	Directors	3 for each array

Antenna Beam Width: About 100 degrees.

3.2.2.7 Power Consumption

It is assumed that 110 V supply would be available for transponder's power requirements so power efficiency of the system is not very critical.

Transmitter	8-10 watts (only during transmission)
Receiver	2-4 watts (continuous)

3.2.2.8 Cost

Ranging between \$1000 to \$2000.

3.2.2.9 Suppliers

a. Cynetics Corporation

P.O. Box 2422

3824 Jet Drive

Rapid City, SD 57709

b. Campbell Scientific

P.O. Box 551

815 W. 1800 N.

Logan, Utah 84321

(801) 750-9555 (Dave Meek)

3.2.3 Weather Stations

A primary application of the ground transponder is the automation of weather/meteorological data gathering. There are presently several companies that distribute weather and stock information to farmers around Iowa. The weather information provided by these companies are very general and cover very large areas in the map grid. More accurate weather information can be obtained at a higher cost than the already expensive standard weather package.

One hundred fully equipped weather stations will provide automated data gathering around the state of Iowa. A Campbell Scientific Universal 10 meter tower with a wide range of sensors will provide the measurement/datalogging instrumentation. A standard IBM 9-pin serial connector will be used to interface with the common power/transponder unit.

These weather stations will provide the farming industry with weather data essential for crop management at a very low cost, if not free. The weather stations will also provide useful meteorological information to the Iowa State University meteorological center.

3.2.3.1 Weather Sensors

A sorting process was made from the "Price List" Campbell Scientific catalog in order to determine which sensors were most adequate for farming purposes and which were most adequate for meteorological studies.

3.2.3.1.1 Farming Sensors

Interviews were conducted with a couple of farmers and with an applications engineer at Campbell Scientific in order to determine which sensors would provide the most useful information for farming purposes. The following Campbell Scientific sensors were selected as "farming" sensors:

- 1) Air temperature and relative humidity sensor.
Model: HMP35C
Description: Vaisala Temp. & RH Probe
- 2) Leaf wetness sensor.
Model: 237L
Description: Wetness sensing grid
- 3) Precipitation Sensor
Model: TE525
Description: Texas Electronics Tipping Bucket Rain Gage
- 4) Soil moisture sensor
Model: 227-L
Description : Soil moisture block

- 5) Soil temperature sensor
Model: 107B-L
Description: Temperature probe
- 6) Solar radiation sensor
Model: LI200S-U
Description: LI-COR Silicon Pyranometer
Note: Most useful solar radiation sensor to farmers
- 7) Wind speed and direction sensor
Model: 05103-U
Description: RM Young wind monitor
Note: Most robust of all the wind sensors.

3.2.3.1.2 Meteorological Sensors

A professor of the ISU Agricultural Department, Dr. Taylor, was interviewed to acquire information on the most useful sensors in meteorological studies. A recommendation was made to add the following sensors to the weather station in order to make it useful for meteorological studies:

- 1) Barometric Pressure
Model: PTA427
Description: Vaisala barometric pressure sensor
- 2) Snow Depth
Model: UDG01
Description: CSC ultrasonic snow depth sensor

3.2.3.2 Datalogger and Other Accessories

Additional peripheral devices and extraneous hardware are needed to make the weather station operational. The following is a list of some of the accessories needed for the weather station:

- 1) Ten meter tower, mounting base, grounding kit, anchors, and guy kit
- 2) CR10 datalogger, extended temperature test, and keyboard / display
- 3) Instrument enclosure and kit.
- 4) Power supply and mounts.
- 5) Datalogger software, interface and cables.

3.2.4 Price List

Sensors and Mounts		
Model	Description	Price
HMP35C	Vaisala Temp. & RH Probe W/5'LD	470.00
UT12VA	Solar radiation shield for HMP35C	210.00
05103-U	RM Young wind monitor	663.00
019ALU	Aluminum crossarm sensor mount	70.00
LI200S-U	LI-COR Pyranometer W/34'LD	220.00
LI2003S	Pyranometer base leveling fixture	44.00
025	Pyranometer crossarm stand	33.00
237-L	Wetness sensing grid W/50'LD	60.50
227-L	Delmhorst soilmoist blk W/25'LD	54.75
TE525	TE rain gage (0.01"/tip) W/25'LD	235.00
107B-L	Water/soil temperature probe W/50'LD	67.00
PTA427	Vaisala barometric pressure sensor	1,045.00
UDG01	CSC ultrasonic snow depth sensor	800.00
Ten Meter Tower		
UT930	Universal 10M tower & adjustable mast	550.00
B18	10M tower concrete mounting base	80.00
UTGND	Grounding kit for 10M tower	30.00

UTGUY	Guy kit for 10M tower	160.00
UTEYE	Eyebolt anchors for 10M tower	30.00
Datalogger and Keyboard		
CR10	Measurement and control module W/CR10WP	1,090.00
XT	Extended temperature test -55 to +85°C	110.00
CR10KD	Keyboard/display	275.00
Enclosure and Power Supply		
ENC	12 x 14" Instrument enclosure	180.00
TN	BKT kit for UT930	25.00
Telecommunications / Data Retrieval		
PC208	Datalogger support software	220.00
SC32A	Optically isolated RS232 interface	145.00
7026	9-pin computer to 25-pin cable	25.00
Total Cost		
Subtotal		6,892.25
Less 5% educational discount		-344.61
Freight & insurance		~160.00
Total		~6,707.64

Table 3.1 Weather Station Cost Estimate

3.2.5 Future Work

One fully equipped weather station should be purchased from Campbell scientific and tested in order to determine how to program the CR10 datalogger and to set data retrieval rates. The weather station can be easily connected to an IBM compatible computer through the 7026 9-pin cable. Tests on the ground transponder should also be made in order to determine the efficiency of the system as automated data retrieval.

3.2.6 References

- (1) Campbell Scientific US. Price List. December 1991.

3.3 Primary CCD Cameras

One of the most common uses of a low Earth orbiting satellite is as a photographic platform. Nothing seems to show off the accomplishment of a successful launch into orbit as well as a picture of the planet as seen from space. This will be the primary purpose of the Charge Coupled Device (CCD) cameras carried on board. They will give direct, unarguable evidence that ISAT-1 has achieved orbit.

Prior to this semester, a trade study was performed to determine which of several models of CCD cameras would be best suited for the project. It recommended the purchase of the Panasonic GP-KR202 full motion model. While no doubt this is a high quality camera, the author does not believe it is the best camera to fly on ISAT-1. Instead it will be shown that the higher resolution Canon Ci-20 still video system may be a better choice.

A Charge Coupled Device (CCD) camera is, put most simply, a photon counting digital camera. The photographic surface is made up of thousands of picture elements (pixels), each of which is an individual photo diode. As photons of light strike each pixel, electrons are knocked from the surface and passed to a series of dynodes each of which amplifies the signal passed to the next. Finally the signal is collected and assigned a value based on the number and the quantum energy (or wavelength) of electrons collected.

3.3.1 Camera Recommendations

Most CCD arrays consist of over 250,000 pixels. As was mentioned above, each pixel is assigned of value based on the photons it receives. For a grey scale camera there are usually 8 bits per pixel (or $2^8 = 256$ levels). Therefore one grey scale picture will require ~ 0.25 MBytes of storage. For a full color picture 24 MBytes are required, (8 for each of red, blue and green), therefore tripling the required storage. Fortunately, digital information has the quality that it can be easily manipulated. If the detector can only discern 300 individual colors from its palette of 16.8 million colors, then the MBytes computer can compress the data

file by using only the color values present. Other techniques such as pixel averaging can also reduce storage space required, but they degrade image resolution.

Even with data compression techniques, it is unlikely that the 9600 MBytes downlink rate will be enough to transmit more than one or two pictures per pass making full motion video transmission impossible even by storing frames for later transmission. This is the primary reason for selecting a still video system rather than the motion video camera. If the latter were chosen, it would require some type of frame grabbing hardware and software to pick out an individual frame. The still camera, however, outputs one frame of digital information directly into the flight computer for either on board processing or downlinking.

With the decision made to use a still camera, the possibility opens up of using a hi resolution color camera that will create enormous data files for each picture taken and then downlinking that image information in segments over several passes, (another benefit of digital technology).

The camera chosen, the Canon Ci-20, was the only industrial grade, still video model for which information was available, Table 3.2.

Number of pixels	380,000 pixels (774Hx488V)
Minimum Sensitivity	20-lux
Power Supply	9V, DC
Power Consumption	5W Max
Operating Temperature	-10 to 45 C (14 to 113 F)
Dimensions (WxHxD)	71x58x90 mm
Weight	Approx. 390 g (13.7 oz)
Minimum exposure time	1 ms

Table 3.2 Technical Specifications for Canon Ci-20

It should be noted that, while the 5W power consumption is more than the Panasonic model, the camera need only be turned for a few minutes around the time of imaging.

It is therefore the author's recommendation that ISAT-1 carry two Canon Ci-20 still video color CCD cameras. The two cameras will share the satellite's 4 inch Maksutov-Cassegrain telescope. In addition to the telescope, one camera will be fitted with additional optics to increase the surface resolution. The optical system is being designed by Gary Cameron of Iowa Scientific Optical.

3.3.2 Data and Resolution

Some preliminary calculations have been made for data storage requirements and image downlink time as well as first estimates of surface resolution without the use of the telescope.

First of all, assuming a full 24 bit per pixel image with the Ci-20's 774Hx488V pixel array, one image will require 1.1 MByte of storage. At a downlink rate of 9600 bits per second, this will take just under 16 minutes which is equivalent to 3-4 satellite passes.

Next, it was assumed that the camera had an angular resolution of 10 micro-radians per pixel (commonly found in text book research but not listed in technical specifications) and the satellite at an orbit altitude of 600 km. Doing this, the surface resolution is calculated to be 6 m without use of the telescope.

There is some concern as to whether the velocity of the satellite will have a large impact on image quality. Again assuming a circular orbit with an altitude of 600 km, the orbit velocity is determined to be 7.56 km/s. Also, it is assumed that the minimum exposure time of 1 ms is used and the satellite is continuously pointing toward the center of the Earth. With this, the satellite covers 6.9 m of ground distance during the exposure time which is about the same as the per pixel resolution. Further calculations will need to be made in designing the telescope.

3.3.3 Future Work

First and foremost, a price needs to be found for the Canon model. If it is deemed too costly, it may be necessary to research other manufacturers. It is still the author's recommendation that a still video system be used. Other specifications which need to be determined are the angular resolution in micro-radians per pixel (from which surface resolution and field of view can be calculated. These will help determine the image quality and assist the design of the telescope.

3.4 GPS Unit

One of the hottest topics in astrodynamics and navigation is the NAVSTAR / Global Positioning System (GPS). The system consists of a constellation of 24 satellites arranged in six orbital planes. Using coded signals from any four satellites, an on board GPS receiver can either calculate its position on orbit at a precise time or downlink the information for computation on the ground. Knowing the precise position of the satellite containing the receiver at various times provides an efficient means of orbit determination which can be monitored by the ISAT Company rather than having an outside source track the satellite.

The author consulted Professor Peter MacDoran from the University of Colorado at Boulder. Mr. MacDoran holds a number of patents in small GPS receiver technology. He was kind enough to send some background information since the author was too ill equipped to even ask the technical questions required in choosing GPS receivers and antennas.

It was learned, however, that GPS units can be made as small as 0.75 grams with a power consumption of 0.3 W. These, though, are the type used on meteorological radiosonde balloons and therefore are probably not suited to the rigors of space flight. The antennas are small domes measuring 2-3 inches in diameter. Mr. MacDoran also gave his assurance that the data stream from the GPS receiver to the ground station is minimal.

For purposes of national security, errors are built into the GPS satellite signal. Therefore, it is only possible to obtain an accuracy on the order of tens of meters which is probably accurate enough for ISAT's purpose. However, using differential techniques (DGPS) it is possible to be accurate to within centimeters.

However, just as with the personal computer industry, big names mean big prices. Fortunately, as with computers, there are smaller companies which produce the same high quality hardware at more reasonable prices. Large companies such as Motorola currently manufacture flight ready hardware. Professor MacDoran has expressed some interest helping the ISAT project through the small business he operates in GPS technology. GPS is certainly the wave of the future and it is the author's strongest recommendation that this study be pursued further.

3.5 Satellite Laser Ranging

Satellite Laser Ranging (SLR), is a relatively new means of tracking earth orbiting satellites. It consists of two main components. First, the satellite must have some sort of reflector on it, either a mirror or a corner cube lens. Second is a ground based laser unit and signal receiver. If it is known approximately where and when the satellite is going to pass over, a laser beam can scan a portion of the sky. As the satellite passes through the scan region, the laser light reflects from the mirrors and returns to a point near the source. By measuring the time required for the light to travel to the satellite and back, its range can be found from which the orbit may be determined. The author is by no means an expert and suggests that future workers do some research to better understand methodology.

Very little had been done prior to this semester in determining the feasibility of SLR on a satellite as small as ISAT-1. In fact all that had been done was obtaining a catalog from Rolyn Optics. Dr. Jeyapalan of the Civil Engineering Department here at Iowa State University was of the opinion that higher quality reflectors would be necessary than those in the catalog.

The advantage of SLR over GPS tracking lies in the non-mechanical nature of the on board equipment. If the electronics in a GPS unit fail, there is no recourse but to have the military track the satellite. With SLR, however, there are no electronics which may fail and absolutely no power consumption on the satellite. The drawback is cost. Constructing a laser and operating it continuously will cost several million dollars which the Iowa Satellite Company can not yet afford. Another possibility is to have some other group such as the University of Texas at Austin, continuously track the satellite. This will still cost several hundreds of thousands of dollars. The third and most financially attractive option is to have it ranged only on occasion. This can be done at an annual cost ~10-30 thousand dollars.

Clearly, much more work needs to be done in determining the feasibility of this segment of the payload. The first thing which needs to be determined is whether the surface area on the bottom of the satellite, (around the telescope opening), is enough to even reflect a discernible signal. Next, it would be beneficial to know how large of a laser would be required to obtain a measurable signal.

3.6 Micro-Meteor Detection

3.6.1 Introduction

In the space environment where the satellite will be orbiting, there will be much space debris. Because of the size of the satellite, micrometeorites, meteors less than 10 mm in length, will be the main concern. In the earth's orbit right now, there are at least 3.5 million particles of this size. A system needs to be devised to detect these micrometeor collisions and locate where on the satellite the particle actually hit. From this an analysis of the possible damage to systems can be ascertained. For additional information on space debris see Appendix C.3.

3.6.2 Design

A catalog from Kistler Piezo-Instrumentation was obtained previously, and it was this catalog where information on accelerometers was found. In order to properly size the accelerometer, the g-force that the accelerometer would see in one microsecond was needed. This is a standard way for comparing accelerometers. By applying the conservation of mass and momentum equations to the satellite and a particle, heading towards each other with equal velocities, the one microsecond impulse was calculated to be about 20,000 g's. This narrowed the process to choosing a high shock accelerometer. Taking into consideration the temperature range involved during orbit as well as size and weight, as preliminary accelerometer was chosen. In addition, charge amplifiers are needed to run the accelerometers. The smallest charge amplifier compatible was chosen. Information about the accelerometer, a Kistler high impedance 8044 accelerometer, Table 3.3; and the charge amplifier, Kistler Model 5037A, Table 3.4.

Size	0.74"length x 0.425" diameter (cylinder)
Mass	7 grams
Power	see Charge Amplifier (Table 3.4)
Cost	530.00 each (24 required)

Table 3.3 Kistler 8044 High Impedance Accelerometer

Size	4.7" wide x 1.3" high x 2.9" deep
Mass	358 grams
Power	0.06 watts
Cost	\$800.00 (4 required)

Table 3.4 Kistler 5037A Multi-Channel Charge Amplifier

The charge amplifier only has six inputs on the three channel model, so for the current configuration of 24 accelerometers, four charge amplifiers would be needed for a total power cost of 0.24 Watts.

3.6.3 Future Work

This has become a preliminary design. Kistler has just come out with a new line of accelerometers and accessories that are designed for the space environment. The people at Kistler strongly recommend looking into these products before using the previously stated ones, and I must agree. The information has not come yet, but they should be smaller and stronger than those in the current catalog. In addition, a configuration for the accelerometers needs to be developed to properly locate the impacts of the micrometeorites. Minimal redundancy should be considered to lower the cost and size constraints of this subsystem.

3.6.4 Appendix / Theory

In order to locate the collision, a series of equations must be solved. A summation of forces in each direction is first performed, as well as a summation of moments in each direction. These equations need to be related to the center of gravity, a known location. From this information and the fact that the force times the distance equals the moment, the impact location may be found. The equations needed are listed below.

$$\sum F_x = m_{cg} u_x \quad \sum F_y = m_{cg} u_y \quad \sum F_z = m_{cg} u_z \quad \text{Eqn. 5.1}$$

$$\sum M_x = I_{xx} w_x - (I_{yy} - I_{zz}) w_y w_z = F_x d_x \quad \text{Eqn. 5.2}$$

$$\sum M_y = I_{yy} w_y - (I_{zz} - I_{xx}) w_z w_x = F_y d_y \quad \text{Eqn. 5.3}$$

$$\sum M_z = I_{zz} w_z - (I_{xx} - I_{yy}) w_x w_y = F_z d_z \quad \text{Eqn. 5.4}$$

KNOWN: $m_{cg}, I_{xx}, I_{yy}, I_{zz}, w_x, w_y, w_z$

MEASURED: u_x, u_y, u_z

UNKNOWN: d_x, d_y, d_z

3.7 Life Sciences

3.7.1 Introduction

The life sciences project was desired to provide students ranging from kindergarten through twelfth grade the opportunity to use an interactive experiment.

3.7.2 Criteria

The experiment needs to be simple. There are two reasons for this, the first being the age level of the students and the second being the limited space available. Dr. Misra of the I.S.U. Seed Sciences Department has been given a copy of all data sent to ISAT dealing with the life sciences project.

3.7.3 Research

Many government agencies were contacted to determine if there was data available to compare with any experimental data we might receive. The application of any experiments that might be adaptable for our satellite were also examined.

After checking with Goddard, Ames, and Johnson Space Centers (all of which deal more with the human size life sciences), the name of Dr. Knott from Kennedy Space Center provided some useful information. Dr. Knott sent summaries of the experiments that he thought might be adaptable to our specifications, but on review, these seemed slightly more difficult than originally thought. Dr. Knott also gave the names of two others that were of great help.

Dr. Brown from Kennedy is a plant physiologist had suggested the use of increased levels of biological activity to determine germination or plant growth. He also suggested the possibility of attaching a transducer to the seed to measure pressure changes in the hull of the seed.

Dr. Guikema from Kansas State University was more than helpful. He sent a package of detailed information which K-State has collected from their experiments. He suggested using one of two methods in which to get the nutrients to the seeds.

1. a rotating well using two cylinders with off-set wells which allow the mixing of seeds and nutrients when aligned.
2. a syringe with multiple chambers separated by a membrane which is punctured when the plunger is depressed, thus mixing the seeds and nutrients, Figures A.3.1 and A.3.2.

After studying the two methods for seed/nutrient mixing, it has been decided to attempt to use the syringe with two chambers. This was decided for multiple reasons, two being that the rotating well was too big to fit the allowable space and the problem of not easily determining the germination or growth of the seed.

The syringe is produced by BioServe Technologies in Boulder, Colorado. When contacted, Dr. Stodieck sent detailed design information describing the syringe. Dr. Stodieck mentioned that they could possibly produce the syringe to our specifications for the cost of the materials, and the sharing of the data collected.

3.7.4 Suggestions

By utilizing the two large payload boxes, two mini-micro-cams (Figure A.1.14), and smaller syringes, it should be possible to have many individual experiments. Two suggestions to depress the plunger are using compressed gas or a spring.

For the compressed gas, the exact amount to obtain the right pressure must be calculated as well as the release mechanism. Also the necessity of a light source will reduce the "experiment" space.

The spring idea would need some release mechanism as well, but that could possibly come from a glow-plug. This glow-plug could also be the light source.

A second experiment could be run using brine shrimp. When the eggs are dried, they are capable of lasting indefinitely. Then when a saline solution is administered, the eggs hatch. The shrimp would have an advantage over the plants if visual data is sent back, because they would swim around for a couple of days to a week unlike the plants which would simply germinate and die in a matter of a couple of days.

3.7.5 Future Work

Dr. Misra has been given a copy of what has been done and what has been suggested and he sounds very enthusiastic about the project. He is going to look into the research that has already been done to make sure that we can accomplish it also.

A working model of the syringe needs to be obtained and specific dimensions need to be determined. Dr. Stodieck said he would try to answer any questions about the syringe or its construction. Dr. Guikema also would be interested in any questions or comments about either the experiments or the hardware we have chosen.

If it is found that these decisions will not work, the next step would be to try to contact Dr. Alstin of Parks Seed Co. in South Carolina. Dr. Alstin helped with LDEF and may possibly have another experiment which could be used on ISAT. Numerous attempts have been made to contact Dr. Alstin, but he is a busy man and very difficult to catch.

3.7.6 Phone Numbers

Dr. Misra	ISU Seed Sciences	(515) 294-6821
Dr. Guikema	Kansas State University	(913) 532-6011
Dr. Stodieck	BioServe Technologies	(303) 492-4010
Dr. Knott	Kennedy Space Center	(407) 853-5142
Dr. Brown	Kennedy Space Center	(407) 853-3165
Dr. Alstin	Parks Seed Co.	(803) 941-4445

Chapter 4

Power Systems

Chapter 4 Power Systems

4.1 Introduction

Designing the power system for a large satellite is a relatively easy task. Unfortunately since our satellite is a small one the power system design is not as simple.

With a limited amount of surface area on which to attach solar cells the power which can be generated is limited. With this low power generation, battery design is a very crucial part of the overall process.

There were many problems encountered in the design process of this system. The rest of this section describes the steps and problems encountered in designing the power system for ISAT-1.

4.2 Solar Array Design

Power analysis for the satellite consisted of first defining the power needed by all of the subsystems onboard. Then with a given solar array size the minimum efficiency of the cells was determined. However this efficiency was out of the range of current technology, so a different approach was used.

First, with a given outer surface area it was decided that the solar cells must cover all of the available area on the satellite. This ruled out earlier concepts where some of the surface area of the satellite was to be used for a video display or sign of some sort. Additional information may be found in Appendix C.4.

The solar cells decided on are to be Gallium Arsenide. These cells are 18% efficient at beginning of life and greater than 15% efficient at end of life. The cells are to be arranged on panels, with each panel consisting of 32 cells in the arrangement shown in Figure 4.1. The satellite will have four panels per side on five of the sides and 2 panels on the sixth side. These two configurations can be seen in Figures 4.2 and 4.3 respectively.

Panel Configuration
32 Cells / Panel
Volatage 32V
Current 225mA
Power 6W
Cell:
GaA
>15%
20 x 40 mm
series

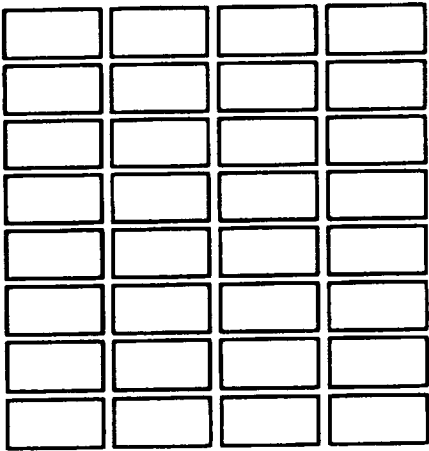


Figure 4.1 Solar Panel Configuration

Main Panel
Configuration
4 Panels / Side

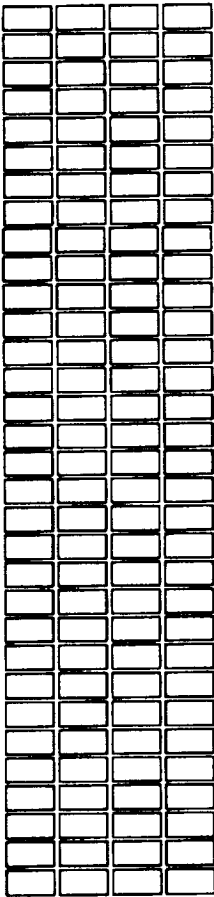
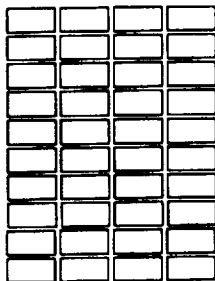


Figure 4.2 Main Solar Array

Bracket Panel
Configuration
2 Panels / Side



?

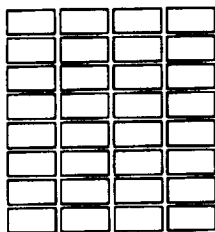


Figure 4.3 Bracket Solar Array

4.3 Power Regulation and Control

The power system for most large satellites runs by drawing power from the solar cells most of the time and only draining the batteries when the amount of power consumption is high. This configuration will not work for a small satellite since the solar array on a small satellite puts out much less power. So the approach used in small satellites is to draw power from the batteries all the time and when the solar array can, it will charge the batteries. This is the setup of the system chosen for ISAT-1.

The output from the solar cells is monitored and controlled by an electronic device that finds the maximum power point for the array and operates the array at

that voltage and current. This power is then used to charge the batteries. After the batteries there is a power regulation unit. This unit has 12, +5, -5 volt outputs and a ground. After the power regulation unit there is a power distribution unit. This unit will be controlled by the flight computer and will control all the power to all the subsystems. It will have current meters on all the outputs to keep track of how much power each subsystem is using. This will allow the flight computer and ground systems to keep track of the power being used and avoid discharging the batteries past their limit.

4.4 Batteries

For a small satellite the selection of batteries is crucial to the performance of the power system. The batteries need to be able to go to low depths of discharge without damaging the battery. They must also be light weight and reliable. There are two kinds of batteries currently being used in satellite applications Nickel-Cadmium and Nickel-Hydrogen batteries. Typical NiCd batteries have an energy density of less than 20 Whr/kg and allow a moderate depth of discharge. NiH₂ batteries have an energy density of 28 Whr/kg and allow for deeper depths of discharge.

The batteries will be drawn upon for power at all times so the battery must be able to go to deep depths of discharge. This consideration and the need to keep weight a minimum necessitated the use of the NiH₂ batteries, even though they are more costly and less common than NiCd batteries (O'Donnell, p. 46).

4.5 Final Configuration

The proposed final configuration is as follows:

Power outputs at 12V, -5V, +5V

Maximum power per orbit of 46.8 Watts

Minimum power per orbit of 33.1 Watts

The amount of power required currently by the subsystems can be seen in Table 4.1. From this it can easily be seen that more power is needed than can be generated if all of the systems are to be running at one time. Fortunately this is not the case. With proper power management there should be enough power to power all the subsystems.

#	Component	Power (W)
1	Structure	0
1	G. G. Boom	0
2	Communications	7
2	Flight Computer	7
2	Large Experiment Module	4
2	Medium Experiment Module	1.5
6	Small Experiment Module	1.1
1	Pitch Torqrod	0.25
1	Roll Torqrod	0.25
1	Yaw Torqrod	2
2	G. G. Boom Tanks	0
1	Earth Radio Frequency Experiment	7
2	CCD Camera	5
1	Telescope	0
6	Battery	0
6	Battery Mount	0
2	Magnetometer	0.04

Table 4.1 Power Budget

4.6 Future Work

To get more power it may be possible to place solar cells under the mounting bracket. This will give an increase of 4 watts in the low power end of the orbit as seen in Figure 4.5. The original power generated without placing solar cells under the mounting bracket may be seen in Figure 4.4. This addition of solar cells and careful power management should allow for all the subsystems to have enough power.

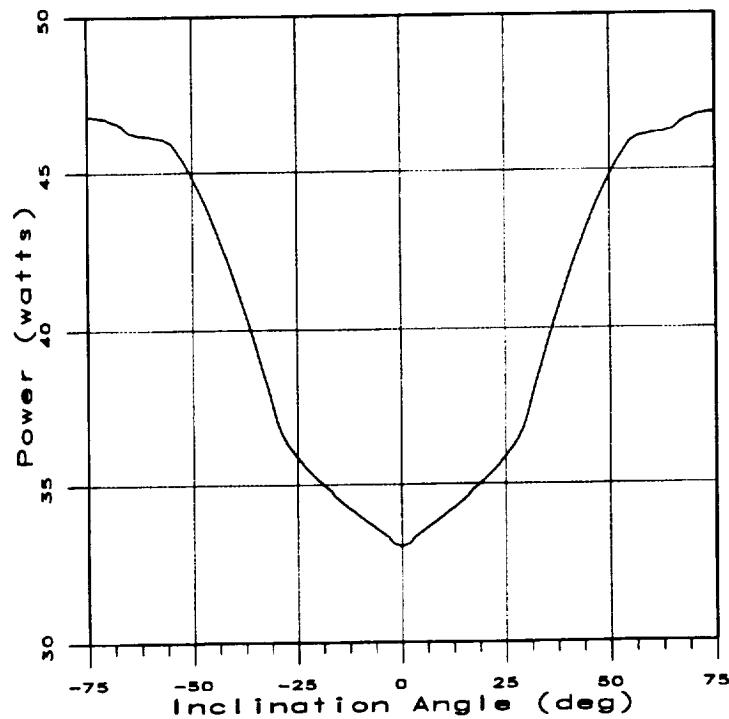


Figure 4.4 Solar Array Power

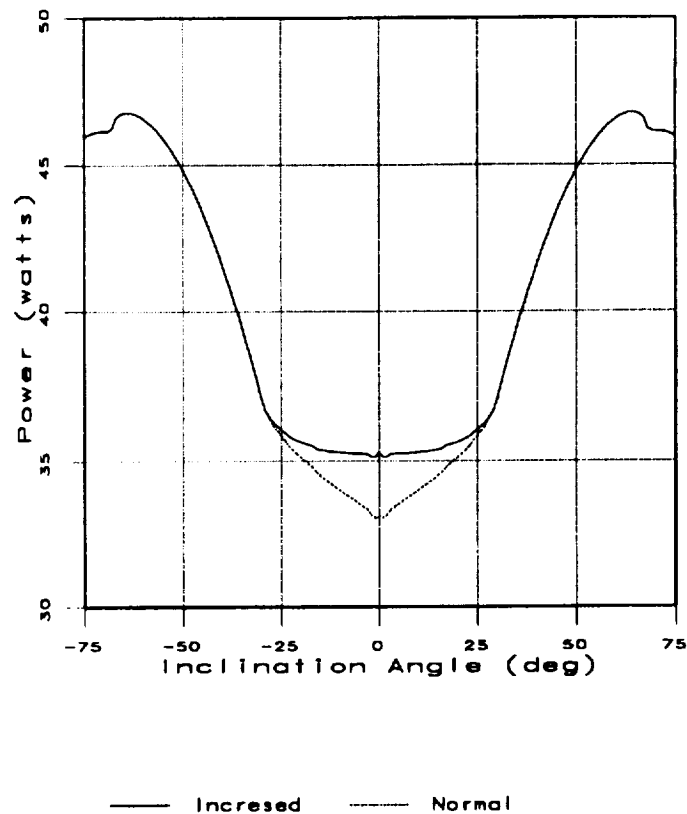


Figure 4.5 Improved Solar Array Power

Things that still need to be examined include the cost of all the components of the system as well as deciding if off the shelf equipment will be used for the power regulator and controllers or if these components will need to be built by the ISAT team.

4.7 References

O'Donnell, Patricia M. "Space Batteries for Mobile Battlefield Power Applications" IEEE AES System Magazine, December 1991.

Chapter 5

Thermal Considerations

Chapter 5 Thermal Considerations

5.1 Introduction

The purpose of a thermal-control subsystem is to maintain all the sections and other subsystems of the spacecraft within their specific temperature limits. This includes all mission phases from takeoff to the death of the satellite. A satellite experiences two major environments direct solar heating and solar eclipse while it is in orbit around the Earth. The thermal environment changes dramatically for these environments and the thermal subsystem must be able to compensate for these changes. Other heat inputs to the spacecraft environment include Earth reflected solar albedo, Earth infrared, and electrical components on board the satellite. For the case of the ISAT-1 all of these have been taken into consideration since they will all affect the thermal environment of the satellite. All of these inputs vary highly with time and it is necessary for the thermal-control system to be able to maintain a working thermal environment for the satellite and all of its components. Although the thermal-control subsystem is one of the most important systems it usually only makes up about two to five percent of the total cost and about the same percentage of the dry weight.

The work that had already been done, was just some basic ideas about what was needed for a thermal-control system for the ISAT-1. Some things considered were such ideas about special coatings for the satellite's skin, heaters, multilayer blankets, and a space radiator that radiate waste heat into deep space. The bulk of the work on the thermal subsystem has been done by the Mechanical Engineering Department at the University of Iowa. This coordination of efforts has turned out to be very educational for both parties of interest. The students at the University of Iowa who were taking a thermal design class got to actually work on something that is going to be reality and it also gave the students in Aerospace Engineering at Iowa State University a chance to learn a little more about thermal aspects in design.

5.2 Background

The spacecraft environment for the satellite was modeled as four types of radiation. These radiations included solar, solar albedo, Earth infrared, and satellite emission and their values are given in Table 5.1. The mean temperature of space was modeled as 4 K. These radiations all affect the thermal environment for ISAT-1 and the thermal-control system to be designed will have to consider all the types and compensate for them. The spacecraft's environment is a harsh one with very transient temperatures and a thermal-control system is the only thing protecting the delicate payloads from this harsh environment.

Spacecraft Environment	-Radiation	Solar $\cong 1430 \text{ W/m}^2$
- Tspace = 4 K		Albedo $\cong 0.3(\text{Solar})$
		Earth Infrared $\cong 230 \text{ W/m}^2$
		Satellite Emission = $\epsilon\sigma AT^4$

Table 5.1 Spacecraft Environment

5.3 Current Work

As a first approximation some assumptions were made to make the first estimates of the thermal environment an easier task. The satellite will be spinning about the centroidal axis so it was modeled as a cylinder. The total internal power generation was estimated to be around 40 watts. The mass of the bus was estimated at 15 kg of aluminum. The transient response was modeled by a mathematical expression relating all radiations presented earlier. A FORTRAN program was then written using these assumptions which were used to specify the thermal environment more accurately and account for the transient response the satellite will encounter. A finite difference method was also added to this program to give even more accurate information about the thermal details. The solar radiation for the program was modeled as a step function having its maximum value stated above when in direct contact and going to zero in eclipse. The albedo radiation had to be modified a little to account for the dusk and dawn periods that

would affect its intensity. It was modeled as constant during daylight, a ramp function during dusk and dawn and zero at eclipse.

Through the use of this program the skin temperature transient response was determined. It was shown that after about five orbits from the initial orbit the fluctuations in the response died out. This left the transient response the satellite would encounter for the rest of its life. High skin temperatures were on the order of about 330 K and low skin temperatures were seen to be about 280 K after the fluctuations died out. A sensitivity analysis was also performed using the program. The length of the satellite was varied while holding all other properties constant. The results showed that for increases in length the mean temperature of the skin went up fairly dramatically.

Once a nominal orbit environment was established it was necessary to upgrade the math model to account for the internal analysis of the satellite. In this upgraded model the satellite was broken up into six sections consisting of top, bottom, sides, and three levels of shelving. Thermal resistance values between these sections were set up in order to isolate certain sections from others. Temperature tolerances between these sections were estimated and the thermal resistance values solved for. Using these values, insulation can be designed and thermal coatings to the exterior and interior of the satellite can be looked at. The insulation and coatings will keep the internal thermal levels within limits set by the other components of the satellite. It is hoped that this will be all that is necessary and that no radiators or heaters will be needed. Since ISAT-1 is a small enough satellite and does not contain such things that require very strict limits this may be possible with the exception of the robotic arm camera which will probably need some sort of special thermal control. Figure 5.1 shows the upgraded math model. A more detailed report may be found in Appendix C.5.

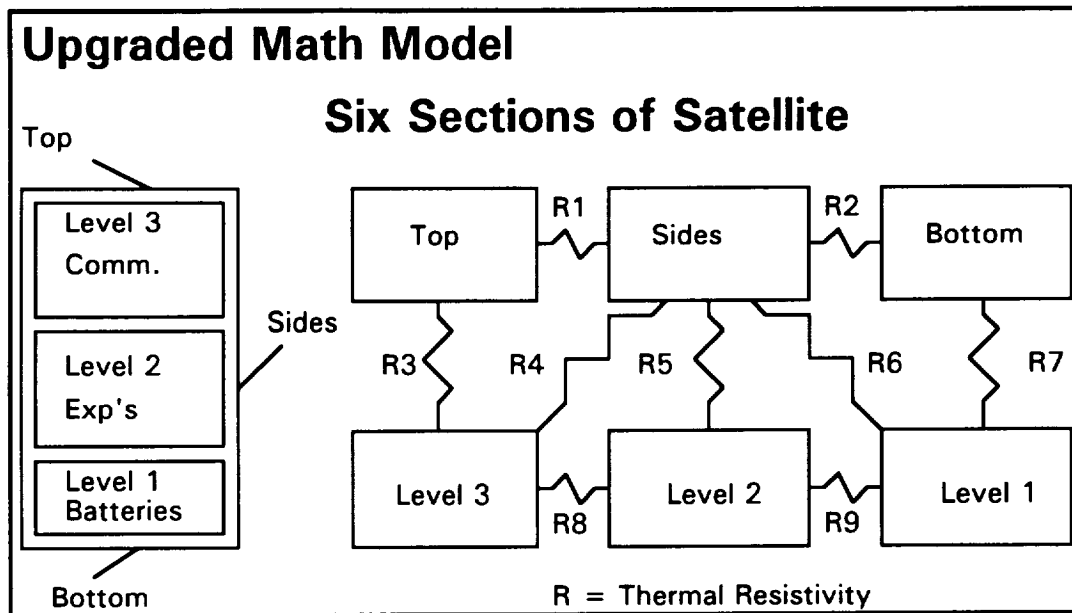


Figure 5.1 Thermal Model

5.4 Future Work

Future work for the thermal-control subsystem depends on the final choices for the other subsystems and payload components. The heat outputs and temperature limits must be specified before design work can continue. When all of these are specified the design work can become more detailed and final decisions made. At the moment not all choices for payload and bus designs are final. The continued upgrading of the math model and results from this model are to be dealt with in future work. Special thermal coatings and insulation are being discussed and will be recommended at a point in which all component choices have been made final. Also, the final decisions about heaters and radiators will depend on these choices. When all choices have been made possible problem areas can be considered.

Another important aspect of the future work will be deciding what thermal tests will have to be performed before launch. A testing schedule will be developed and testing cycles decided upon. These tests will include component-level tests and system-level tests. The testing process is very important to the dependability of the thermal-control system. For example, multilayer blankets are

affected quite considerably by overlapping at joints and their thermal performance degrades quite severely. They are also very sensitive to being handled and installed since smashing them could ruin their thermal productivity. Also, test times in vacuum chambers must be long enough for the air trapped within the multilayer blankets to vent. Otherwise the blanket will not perform as it would in space. This is why the testing of the thermal system is so very important. If anything gets by the testing the whole satellite project could be jeopardized.

5.5 References

Larson, Wiley J., Wertz, James R. 1992.. *Space Mission Analysis and Design*.
Second Edition. Kluwer Academic Publishers. 409-430.

University of Iowa Participants

Butler, P. Barry, NASA, UI, 319-335-5672

Prall, David, student, UI, 319-339-7727

Schneider, Matt, student, UI, 319-338-4234

Smith, Ted, professor, UI, 319-335-5680

Skrbich, Mike, student, UI, 319-339-4901

Iowa State University Participants

Reinders, Brent, student, ISU, 515-292-0532

Chapter 6

Attitude and Control

Chapter 6 Attitude and Control

6.1 Introduction

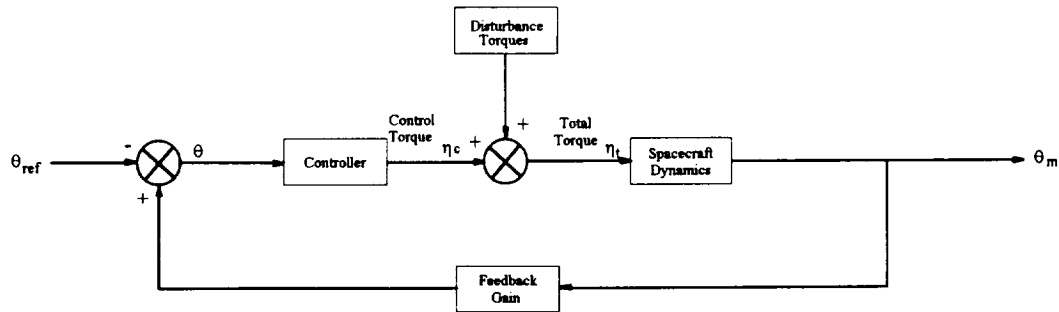
The attitude determination and control subsystem stabilizes the vehicle in the correct orientation for the mission. This requires both accurate knowledge of the spacecraft attitude and 3-axis control to reject disturbances.

Sensors use external references such as the sun, the earth's horizon, the earth's magnetic field direction and the stars to correctly determine the vehicle's attitude. Control techniques may include: gravity-gradient control, momentum wheels, reaction wheels, control moment gyros, thrusters, or magnetic torquers.

6.2 Overview For ISAT-1

Three scenarios must be considered for the control of ISAT-1. These are: despin, normal operation and inverted orientation. Despin includes the damping of initial spin rates from orbit insertion. For normal operation, a more precise attitude is required and disturbances must be rejected. Inverted orientation is a special problem which must be addressed due to the use of the gravity-gradient boom.

A block diagram for ISAT-1 is shown in Figure 6.1. The disturbance torque's which may be encountered include: third body effects, oblateness, hits, solar radiation, residual dipole and atmospheric drag. All disturbances except the last two are considered negligible for this vehicle. For each of the three scenarios mentioned, a controller and feedback gain must be determined.



Disturbance Torques:

- 1) hits
- 2) oblateness
- 3) 3rd body effects
- 4) solar radiation
- 5) residual dipole
- 6) atmospheric drag

For each scenario the controller and the feedback gain must be designed.

Figure 6.1 Control Block Diagram

6.3 Gravity-Gradient Control

For ISAT-1 a passive control technique called gravity-gradient control will be utilized. This method requires neither power supply nor attitude sensors. Instead, the inertial properties of the vehicle will determine its stability.

A body subjected to gravitational force naturally aligns its axis of minimum inertia along that line of force. Therefore, by attaching a boom onto ISAT-1, its inertial properties can be altered so that it is inherently stable in an earth-facing attitude.

It has been found in historical data that $s_x = .98$ provides the desired stability, s_x is defined in Eqn. 6.1. On the other hand, $s_y = \frac{1}{3}$ (Eqn. 6.2) should be avoided because it would cause resonance.

$$s_x = \frac{I_y - I_z}{I_x} \quad \text{Eqn. 6.1}$$

$$s_y = \frac{I_x - I_z}{I_y} \quad \text{Eqn. 6.2}$$

6.4 Magnetic Torquers

Since gravity-gradient control causes oscillations and does not stabilize the local yaw axis, another method of control is needed.

Magnetic torquers provide an inexpensive method of three-axis control with no moving parts and relatively low power consumption. Magnetic coils on rods generate dipoles and produce torque proportional to the Earth's changing magnetic field.

6.5 Sensors

The attitude sensors chosen for ISAT-1 are a sun sensor and a magnetometer. These are light-weight, inexpensive, require little power, and are sufficiently accurate for ISAT-1's mission. Sun sensors detect the angle of incident light to determine orientation. Magnetometers measure the direction and magnitude of the Earth's magnetic field and are used in conjunction with the torque rods.

6.6 Despin Model

In order to solve the problem of attitude control for ISAT-1 a computer simulation has been designed. For the despin scenario, the satellite has been modeled as a 50 kg homogeneous mass before boom deployment. The orbit is assumed circular with a semi-major axis of 7178 km and an inclination of 50 degrees.

A bang-bang control law was used to model the torque rods on ISAT-1. The torque (T) applied to the vehicle is equal to the saturation moment (m_{sat})

created by the torque rod and is applied in the opposite direction of the changing magnetic field ($\frac{db}{dt}$).

$$T = -m_{\text{sat}} * \text{sign} \left(\frac{db}{dt} \right) \quad \text{Eqn. 6.3}$$

A spherical harmonic Earth magnetic field was used and the saturation dipole of the torque rod was set a $1.0 \text{ A}\cdot\text{m}^2$ this creates a maximum possible torque of $4.3(10^{-5}) \text{ N}\cdot\text{m}$ on the vehicle.

Using rigid body dynamics and two-body orbital mechanics with the one primary being Earth, test runs were performed. The program is included in Appendix B.6. An initial spin rate was given to each body axis and the despin control history was shown graphically in Figure 6.2. For each of these cases, all disturbance torque's were neglected and perfect attitude knowledge was assumed. As can be seen in Figure 6.2, reasonable initial spin rates can be brought under control after only a couple of orbits.

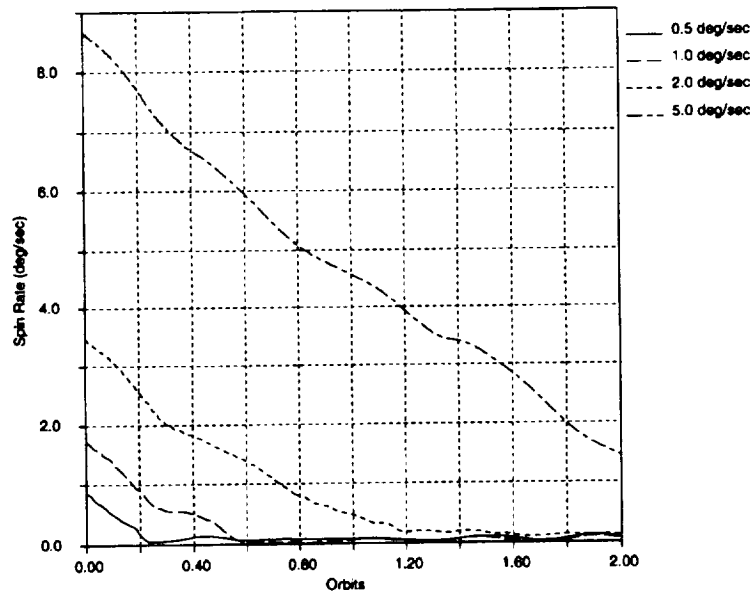


Figure 6.2 Despin Control

6.7 Capture

After the despin has been initiated, the next important procedure is to determine exactly when the gravity gradient boom should be deployed. The object is to have the right conditions so that the satellite ends up in the proper orientation with respect to the Earth. In order to accomplish this, a relationship between pitch rate (\dot{q}) and pitch attitude (q) needed to be found. Equation Eqn. 6.4 is the resulting equation from the original pitch equation and is the relation between \dot{q} and (q).

$$\dot{a}^2 = \frac{\dot{q}^2}{3 \omega_o^2 s_{y0}} + \sin^2(q_0) \quad \text{Eqn. 6.4}$$

Assuming that there are no external torque's acting on the body, and if (\dot{a}) is held constant and (\dot{q}) and (q) are varied from $\frac{-3p}{2}$ to $\frac{3p}{2}$. From this information, a stable range for boom deployment can be found around (q) equal to 0 and $a < 1$. The corresponding (\dot{q}) quantities can then be determined.

6.8 Future Work

Since the control of ISAT-1 is completely dependent upon attitude knowledge, the modeling of sensors is very important. A complete understanding of the sensors is needed, and a method of filtering the data must be developed.

The effects of residual dipoles and atmospheric drag must be added to the computer simulation and analysis. It must first be determined what will be the residual dipoles that are caused by internal electronics in the satellite. This required knowledge of all payload systems.

The scenarios of normal operation and inverted orientation must also be analyzed. In normal operation, a method of dampening precession and nutation must be found. For inverted orientation, the problem of flipping the spacecraft back to a normal orientation must be solved.

Chapter 7

Ground Systems and Operations

Chapter 7 Ground Systems and Operations

7.1 Introduction

Ground Systems and Operations cover a wide field of study. It deals with the planing of all the phases of the Iowa Satellite Project, starting from the conceptual design until ISAT-1 plummets into Earth's atmosphere. First, it is responsible for the compilation of detailed time lines, both overall and a more detailed test schedule. Second, systems for the ground station will be selected. Finally, flight operations and procedures must be decided on.

7.2 Time Lines

The time lines will provide a basis for an orderly development of ISAT-1. Time lines must be made for many aspects of the project to give designers and operators a sense of direction. This spring semester, the focus of operations was on a revision of the overall and more detailed test time lines. First, the overall time line maps the phases of development from preliminary design to sustained operations. Second, the test time line was revised to reflect changes in the overall time line and more detailed input from design groups.

7.2.1 Overall Time Line

Now knowing that it is possible to have a late 1995 launch it was necessary to revise the overall schedule (since the Summer 1992 report was based on a late 1994 launch).

7.2.1.1 Preliminary Design

This phase has primarily been completed. This includes the development of the overall concept of the satellite. The payload packages were selected. The communication frequencies have been requested.

7.2.1.2 Detailed Design

This semester has been focussed on this phase. Knowing the conceptual design, this phase concerned the design and selection of hardware that will compose each subsystem. Specific subsystem configurations will be drawn, down to the wiring. Ground system configuration will also be decided .

7.2.1.3 Construction

This phase will be the actual fabrication of the satellite. Components will be purchased and made for each subsystem and hardware for the ground station will be purchased.

7.2.1.4 Testing

Every subsystem, system and component must be tested to determine if that unit will operate in orbit. There will be four levels of testing.

7.2.1.4.1 Proof of Concept

The proof of concept test will determine if a certain subsystem will be feasible and will provide the data that is desired. For example, the Meteoroid Location package could utilize a computer code. This code would be used to determine the position of particle impact from acceleration values in certain locations.

7.2.1.4.2 Acceptance Tests

From the detailed design, a list of components that comprise each subsystem, will have been compiled. This detail design will govern the operating parameters of each component, while the acceptance test will determine if these parameters are met. Again for the Meteoroid package, the accelerometers should be tested to determine if they meet the factory specifications (since they will be purchased).

7.2.1.4.3 Integrated Tests

The Integrated tests will consist of connecting all the components of a particular subsystem and determine if they work together. A possible test for the Meteoroid location package could utilize the mockup of the satellite. A number of accelerometers will be attached to the structure and connected to an outside computer at which time a simulated meteoroid impact will be conducted (possibly a bb gun). Then the computer will record and process the acceleration values and determine the location.

7.2.1.4.4 End-to-End Tests

Finally, end-to-end testing will be done. This will test the satellite, which is fabricated and put together as it would be in orbit, to determine if all the subsystems will work together (even the communications link will be used). Here, the meteoroid location payload will be operated in tandem with the other subsystems in order to determine if this payload will operate adversely in conjunction with its neighboring subsystems (such as electromagnetic interference).

7.2.1.5 Training

This phase will provide future operators working knowledge of the satellite. This will be a combination of lecture, workbook and hands on. For example, the operators must know how to operate the robotic arm without affecting the satellite's stability.

7.2.1.6 Shipment

This phase will transport the satellite to the launch site. A method of transportation must be found that will not damage the satellite.

7.2.1.7 Pre-launch Tests

These tests will confirm the operational status of the satellite. This phase will determine if damage was incurred to the satellite during transportation or some other source. This will also include on pad tests, if at all possible.

7.2.1.8 Launch

Here the satellite's health will be observed during launch.

7.2.1.9 Initial Ops

Following launch, certain events must be performed to make the satellite operational. Also a series of tests must be completed to determine if the satellite is operating properly.

7.2.1.10 Sustained Ops

Sustained operations will begin when the satellite has passed all tests. This is the scientific data collection phase. Operation time lines, that outline step by step procedures, must be written for periods of time, such as a month.

7.2.2 Detailed Test Time Line

The entire satellite must be tested. That includes tests for all components, down to the last wire. Thus it is necessary to develop a time line that will orderly layout the testing periods. This will make it possible to provide certain tests with information collected during a previous test. The following list of tests has a format consisting of three parts, the begin and completion dates, equipment used to conduct the test, and a brief description.

7.2.2.1 Structures Group

7.2.2.1.1 Buckling Analysis of Gravity-Gradient Boom

Begin:	Jan 93
Completion:	May 93
Equipment:	ANSYS, Workstation

Determine the required boom sizing for the boom to withstand space environment.

7.2.2.1.2. Boom Thermal Bending

Begin:	Jan 93
Completion:	May 93
Equipment:	ANSYS, Workstation

Determine the bending properties of the boom and if necessary change the boom's design characteristics to operate safely with respect to thermal fluctuations.

7.2.2.1.3 Outgassing Properties of Composite Materials

Begin: Jan 93
Completion: May 93
Equipment: Vacuum chamber, scale, MTS, etc...

This experiment exposes composite material to simulated space environment. Changes in weight and material properties should be recorded.

7.2.2.1.4 Material Property Tests

Begin: Jan 93
Completion: May 93
Equipment: MTS and computer/software package

Sufficient data should be taken so that accurate material properties can be used in the finite element modeling.

7.2.2.1.5 Finite Element Analysis

Begin: Jan 93
Completion: May 93
Equipment: ANSYS, Workstation

This test will verify that the boom will withstand the forces it will experience during flight. This test should also simulate the launch phase.

7.2.2.1.6 Release Mechanism Verification

Begin: June 93
Completion: Dec 93
Equipment: Command equipment

Verify consistent operation.

7.2.2.1.7 Dynamic Testing

Begin: Sep 93
Completion: Dec 93
Equipment: Shaker, Accelerometers, computer, etc...

Vibrational testing should simulate both launch loads and transportation (to launch site) loads. This experiment should be the combined effort of both the structures and integration teams.

7.2.2.1.8 Non-Destructive Evaluation

Begin: Sep 93
Completion: May 94
Equipment: Dependent on type of material

Verify that composites and/or aluminum is free of defects.

7.2.2.1.9 Strain Gauge Check

Begin: Sep 93
Completion: Dec 93
Equipment: Ohm meter, solder, wire, shaker

This experiment will determine the status of the strain gauge and how they will react to vibrational loads.

7.2.2.1.10 Integrated Structure Tests

Begin:	Jan 94
Completion:	Aug 94
Equipment:	Shaker, accelerometers, computer, etc (foam core components)

The structure will be tested to withstand launch and transportation loads. The structure will include the strain gauges and components represented by foam core materials (this will determine if all the components will fit). The test will also determine the data collection of the strain gauges.

7.2.2.2 Power Group

7.2.2.2.1 Solar Array Output

Begin:	Sep 93
Completion:	Dec 93
Equipment:	Volt & Amp meters, Simulated sun source

This experiment will determine the status of each individual solar cell.

7.2.2.2.2 Battery Capacity

Begin:	Sep 93
Completion:	Dec 93
Equipment:	Volt & Amp meters, load

Verify the battery parameters.

7.2.2.2.3 Power Regulation Device

Begin:	Sep 93
Completion:	Dec 93
Equipment:	Volt & Amp meters, power supplier, power storage device, and load

This test will determine the operational status of the power regulation device.

7.2.2.2.4 Internal Meter Check

Begin:	Nov 93
Completion:	Dec 93
Equipment:	Volt & Amp meters, loads, power supply and storage devices

This test will verify the subsystem, that determines the status of the entire power system, is working.

7.2.2.2.5 Integrated Power System Tests

Begin:	Jan 94
Completion:	May 94
Equipment:	Volt & Amp meters, load, simulated Sun

Determine the characteristics of the entire power system. Verify the charging of the batteries directly through the solar array and so forth.

7.2.2.3 Attitude Control and Determination Subsystem

7.2.2.3.1 Gravity-Gradient Simulations

Begin:	Jan 93
Completion:	June 93
Equipment:	Workstation/Computer, Code

The code developed for this simulation should be general, allowing for the user to adjust the satellite's properties (distribution of mass).

7.2.2.3.3 Attitude Sensors

Begin:	Sept 93
Completion:	Oct 93
Equipment:	?

Determine if the sensors operate within manufactures' parameters.

7.2.2.3.4 Torque Rod Test

Begin:	Oct 93
Completion:	Dec 93
Equipment:	?

7.2.2.3.5 Integrated ADC Test

Begin:	Jan 94
Completion:	May 94
Equipment:	Mockup satellite, 3 dimensional rotating stand, simulated forces.

This test will place a mockup version of ISAT-1 in a frame that is allowed to rotate in 3 dimensional space. The mockup will have simulated forces applied to it in order to determine the status of the torque rods (This test may not be valid, since the test is on the Earth's surface).

7.2.2.4 Payload

7.2.2.4.1 Accelerometer Data

Begin:	Jan 93
Completion:	May 93
Equipment:	Mock-up, accelerometers, computer, code, bb gun.

Using computer simulation, determine if the combination of accelerometer data can be triangulated to determine the locations of impacts. Develop required software.

7.2.2.4.2 Laser Reflectors

Begin:	Jan 93
Completion:	Aug 93
Equipment:	?

Determine the feasibility of the laser ranging experiment.

7.2.2.4.3 Plant Experiment

Begin:	Jan 93
Completion:	Aug 93
Equipment:	?

Determine the feasibility of the plant experiment.

7.2.2.4.4 Robotic Arm

Begin:	Jan 93
Completion:	Aug 93
Equipment:	Computer, Simulation Code

In conjunction with attitude group, determine the stability of the entire satellite when the arm is functioning. Also, determine, via computer simulation, the envelope of the arm.

7.2.2.4.5 Radiation Detector

Begin:	Sep 93
Completion:	Dec 93
Equipment:	Radiation source, Reference detector, computer.

7.2.2.4.6 Acoustic Sensor/Micrometeoroid Detector

Begin: Sep 93
Completion: Dec 93
Equipment: ?

This is a follow up on the proof of concept test concerning the accelerometers.

7.2.2.4.7 CCD Camera

Begin: Sep 93
Completion: May 94
Equipment: Thermal Chamber, shaker, ?

Determine the survivability of the CCD Camera.

7.2.2.4.8 Telescope

Begin: Sep 93
Completion: May 94
Equipment: ?

Determine if the telescope can switch power settings.

7.2.2.4.9 Robotic Arm Mock-Up

Begin: Sep 93
Completion: Dec 93
Equipment: ISAT-1 mock-up and command software

Using command software, can the robotic arm be controlled? Can failures be corrected.

7.2.2.4.10 Video Display

Begin: Nov 93
Completion: May 94
Equipment: Thermal chamber, Volt & Amp meters, video camera

Determine the characteristics of the robotic arm camera.

7.2.2.4.11 Status Check Subsystem Tests

Begin: June 94
Completion: Aug 94
Equipment: ?

Tests must be done to determine the operational status of each experiment package. Certain failure scenarios should be listed and then determine how each scenario can be concluded upon, via sensors.

7.2.2.4.12 Meteoroid Location System Tests

Begin: Jan 94
Completion: May 94
Equipment: ISAT-1 mockup, accelerometers, computer, code, micrometeoroid equivalents, Volt and amp meters.
Check if the system still functions with the predicted momentum transfers. Measure power consumption.

7.2.2.4.13 Telescope and CCD Camera Combo

Begin: Apr 94
Completion: Aug 94
Equipment: Telescope, CCD camera, visual reference, computer.

Determine the compatibility of the two units and the operational reliability using commands via a computer.

7.2.2.5 Thermal Control System

7.2.2.5.1 Thermal Couple Check

Begin: Sep 93
Completion: May 94
Equipment: Volt meters, thermal chamber

7.2.2.5.2 Thermal Determination Subsystem

Begin: Sep 93
Completion: May 94
Equipment: ?

Determine the operational status of the thermal detective devices in order to determine the thermal environment.

7.2.2.5.3 Integrated Thermal Control Test

Begin: June 94
Completion: Aug 94
Equipment: Thermal chamber, IR source, etc...

7.2.2.6 Communications

This topic can be found in the Summer 1992 report. This section was not revised due to insufficient knowledge of the subject.

7.3 Data Management

Data management is concerned with how much data in bits of information each subsystem will be allowed to generate. To determine this, it was necessary to discuss the overall configuration of the satellite and ground system, and the orbital characteristics.

7.3.1 Satellite & Ground System Configuration

The satellite configuration has three levels. The payload and health check systems are represented by the first level. On the second level are the two flight computers and data storage devices. On the final level are the two communication packages. These comm units transmit to the ground antenna at Boone which is connected to a networking system. This network consists of the primary computer, tracking computer, data storage computer and filtering computer.

The data created by the subsystems, health checks, and monitoring systems will be routed to one of the flight computers via a serial junction. The data will either be directly sent to the next serial junction or stored in the memory of the flight computer to be sent later. The second serial junction will route the data to a comm unit for transmission back to Boone. There is an inherent limiting factor in these communication packages. They operate at 9600 baud (9600 bits per second).

7.3.2 Orbital Characteristics

Since the rate of the transmitter is known, the next step is to determine the amount of time we have to transmit. This is inherent in the satellite's orbital characteristics.

A selection of orbits have been selected for analysis. Each orbit has two sets of data. First is the line of sight data, when a person at Boone can observe the satellite. The second set of data is the communications envelope. Any time the satellite is within this envelope, a communications link can be established. This data gives the time of pass, pass duration, time since last pass, and total data transmitted during the duration. For example, taking an orbit of 800 km in altitude and inclination of 50 degrees a pattern can be seen. Generally the passes come in sets of 5 and 4 with a hour and three quarters time interval, see Figure A.7.1 and Table A.7.1. Each pass ranges up to 6 minutes in length. Between each set of passes there is roughly a 17 hour period. The main item to note, is that the orbit's durations are not constant throughout the set of passes. Each pass must be dealt with individually.

7.3.3 Data Rate

With this overall picture it was possible to determine when and how long the satellite is in communications range. The data displays the time for each pass over an arbitrary six day week. With the duration of each pass and the rate of the transmitter known, it is possible to create a basic data transmitting schedule.

An example of data transmitting is given below,

$$9600 \text{ baud} = 9600 \text{ bits per second}$$

$$1 \text{ Byte} = 8 \text{ bits}$$

$$9600 \text{ baud} = 1200 \text{ bytes per second}$$

For 8 minutes of transmitting (which corresponds to the average maximum duration).

576,000 bytes transmitted

(.576 Megabytes)

This corresponds to about a little over a third of the memory on a 3.5 inch High Density floppy (IBM formatted 1.44MB). Thus, it will be necessary to limit the data flow of each subsystem.

7.3.4 Data Rationing

Having only a limited amount of data that can be transfer it is necessary to delegated sizes of data creation to individual subsystems. Each orbit sets an Xi amount of data that can be transmitted during each pass i. This amount can be calculated from the Orbital Appendix (duration of the pass times the data rate). For example, the 45th pass for the 800 km altitude, 60 degree inclined orbit has a duration of 3.77 minutes. This corresponds to a total data transmission of 4.35 Mbits (having two communication units operating at 7600 baud). Each subsystem is given a portion of this amount to transmit down to the Boone station. Each group must utilize this transmitted data in an useful manner.

7.3.4.1 Data Receiving Groups

The data will be divided into five groups. Command and Control, Structures, Attitude Determination and Control, and Power will be portioned a minimal (enough to be useful) amount of data. This amount of data will be a set value; every pass will consist of this size of data transmission. The Payload group will have the left over amount. This amount must be divided among each payload subsystem (decided by the payload group and specific event).

7.3.4.2 Command and Control

The data collected for Command and Control is basically confirmation requests. During operations, certain events are initiated. If an event is initiated prematurely, by accident, etc it may be detrimental to the satellite. Thus, each command will be followed by a transmitted confirmation statement. For example, an operator types in the commands to extend the boom. The satellite receives this uplink and automatically transmits a confirmation statement where the operator will reply either yes or no (whatever the conditions are at that time).

7.3.4.3 Structures

Multiple strain gauges and other devices will be attached to the structure in various locations (locations that are important). These measuring devices will create data that will be processed into structural status report.

7.3.4.4 Attitude Determination and Control

ADC will be given a certain amount of data. This amount must be manipulated in a fashion so that the attitude of the satellite can be determined in the least amount of space. This will require analysis in what type of measurements are to be made and how often they are made.

7.3.4.5 Power

The power group will be delegated an amount of data transmission to determine the operational status of the system. Voltage and current checks must be done in various areas of the subsystem to determine if each component is working properly (solar cells, batteries, regulation device, etc.).

7.3.4.6 Payload

The payload group will receive any left over amount of data that the other subsystems do not use. This data must be delegated by the group among the various payload packages.

7.3.4.7 Data numbers

As stated earlier, Command and Control, Structure, ADC, and Power will be given a set amount of data transfer per pass. Each group will receive 10,000 bits of data per pass. Depending on the duration of the pass, the payload will pick up the rest of the data transfer.

7.4 Personnel

It will be necessary to hire personnel to operate the satellite. The length of time for each set of passes is roughly 7 hours for 5 pass set (set type A) or 5.3 hours for 4 pass set (set type B). If we allow an hour and a half prior to the initial pass and an hour of logging out (including composing a report concerning the pass set), operators will work a 10 or 7 hour shift. One factor to note, is the shifts do not always come at desirable time (such as a 7 to 5 job). Some shifts may start a midnight while others may start in late afternoon. Whatever the case may be, the Boone station must be manned 17 hours out of a 48 hour period.

7.5 Ground Systems Selection

7.5.1 Purpose

The final area focussed on is ground systems selection. This is the selection of hardware for the Boone ground station. The hardware must operate properly; communicate with and track the satellite, store data, monitor health status, and initiate events.

7.5.2 Ground Station Layout

The main room contains the control three-bay, educational desks, equipment rack and secretary's area. The storage room will be used for storage and as a snack room for operators, it will also contain the large screen TV. The executive room will consist of the meeting table and the engineers work space. As of March 15, 1993 the equipment that will be purchased is shown in Table 7.1. A schematic diagram of the communications system is shown in Figure A.7.2.

80486/50 Control Computer 20" SVGA Color Monitor 80386SX/20 Control Computer 20" SVGA Color Display 80386/33 Data Archive Computer 17" SVGA Color Display 80486/50 Engineer's Computer 20" SVGA Color Display Macintosh 2vx Education Computer 16" SXVGA Color Display	Laserjet 4 Deskjet 550C DraftPro DXL NP2020 Copier Scanjet IIc Laserjet FAX
Ferrups UPS RE1.8KVA Citadel Power Conditioner CLT1001RTW	

Table 7.1 Ground Equipment

The furniture will be from Rockwell and AMCO. The equipment rack, three-bay and some of the furniture will be purchased from AMCO. Rockwell will donate some furniture and possible some hardware.

7.5.3 Network

The computers at the Boone Ground Station will be consolidated in a networking scheme. This will provide faster processing of data by having the individual computers parallel process their designated data. For example, the data for tracking the satellite will be directed to the tracking computer where it will be processed, while at the same time, the health status data will be sent to its computer for processing. Several options have been looked at: UNIX, Novell, Windows NT. At the time of this report, the process of choosing has just started. The Windows NT, though, looks the most promising. It is however not on the market. It is believed that it is in the beta version phase and that a copy will be made for ISAT testing usage.

7.6 Future Work

Only a portion of Ground systems and Operations has been covered to date. The following list is certain areas that need to be determined,

- 1.) A final detailed test time line
- 2.) A revision of the data management
 - Need a final orbit
- 3.) Final ground systems selection
- 4.) Event Scenarios
 - Problem solving procedures
- 5.) Personnel Management

Study of the above areas will be attempted, time permitting.

7.7 References

Iowa Satellite Project. Summer 1992 Report. Ames: ISU, 1992

Chapter 8

Communication System

Chapter 8 Communication System

8.1 Summary/Overview

Functionally the communication system of the satellite can be divided into two categories. The main comm system which is a part of the satellite bus and the experimental comm system which is a part of the satellite payload. For the purpose of identification the first one is designated as COMM-1, Figure 8.1, while the other as COMM-2, Figure 8.2.

Practically the two systems are not totally independent. They share some of the common system elements and also one of them (COMM-2) acts as a standby to the other (COMM-1).

In the later part of this section, design considerations for COMM-1 and COMM-2 are presented.

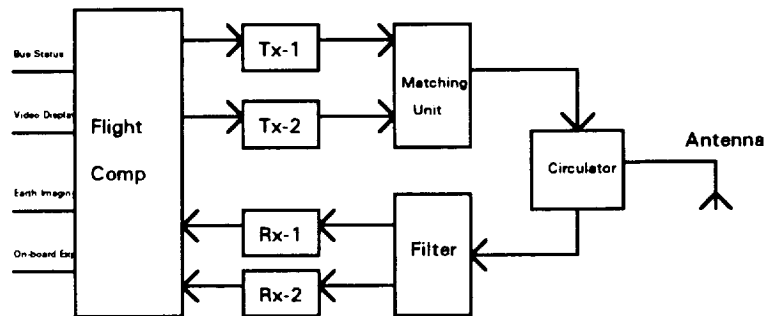


Figure 8.1 COMM-1 Block Diagram

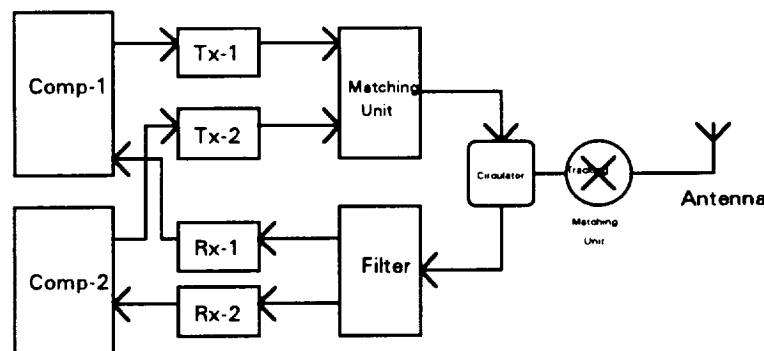


Figure 8.2 COMM-2 Block Diagram

8.2 COMM-1

8.2.1 Purpose

This would be the main communication link between satellite and ground station. All the Command and Control data would be handled through this link. Additionally, following other types of data would also be transferred using this link:

Uplink:

- a. Picture display data.
- b. Onboard computer software update data.

Downlink:

- a. Bus health status.
- b. On-board experiments data.
- c. Earth imaging data.
- d. Ground sensors data.

8.2.2 Data Rate / Bandwidth Calculations

Average number of passes per day	4
Average contact time per pass	10 min. or 600 sec

Uplink:

- Data to be uplinked:
- a. Command and Control data
 - b. Software Update
 - c. Picture display data

Assumption of data rate	less than 10k bps
Bandwidth requirement	10k Hz

Downlink:

- Data to be downlinked:
- a. Command and Control data
 - b. Bus status report
 - c. Earth imaging data
 - d. Ground sensors data

Assumption of data rates:

Ground sensors	10k bps
All other	10k bps
Total	20k bps
Bandwidth requirement	20k Hz

8.2.3 Frequency Selection

The recent World Administrative Radio Conference (WARC) in Spain allocated new frequencies for small satellite systems. These are:

<u>Frequency Band</u>	<u>Purpose</u>
137.000---137.025 MHz	Primary
137.175---137.825 MHz	
148.000---149.900 MHz	
400.150---401.000 MHz	
137.025---137.175 MHz	Secondary
137.825---138.000 MHz	

Although the ultimate selection of frequency is subject to the availability of a particular frequency band and approval by the FCC, following are the relative merits and demerits of above mentioned frequency bands:

- At higher frequencies the available bandwidth is larger
- At higher frequencies antenna sizes become smaller, thus easier to manage.
- Free space loss is higher at higher frequencies.
- Doppler shift is more significant at higher frequencies than lower ones.

Considering all these factors and our system requirement it appears that the choice of higher frequency band would be more suitable for our satellite.

8.2.4 Link Calculations

Following link equation is used for link calculations:

$$(SNR) = (EIRP) + (Gr/Ts) - (FSL) + (228.6)$$

where

(SNR) = signal to noise ratio

(EIRP) = equivalent isotropic radiated power

(Gr) = receiver antenna gain

(Ts) = equivalent receiver noise temperature

(FSL) = free space loss

Link calculations are done at two different frequencies.

$$f_1 = 150 \text{ MHz wavelength}_1 = w_1 = 2.0 \text{ m}$$

$$f_2 = 450 \text{ MHz wavelength}_2 = w_2 = .67 \text{ m}$$

Minimum antenna angle above horizon = 15 degrees

Orbital altitude = 550 km

Maximum distance between satellite and ground station

$$R = 550 / \sin 15 \text{ or } 2125 \text{ km}$$

$$(FSL)_1 = 142 \text{ db}$$

$$(FSL)_2 = 152 \text{ db}$$

Downlink calculations:

$$(T_s) = 1000 \text{ K}$$

$$(G_r) = 10$$

$$(G_r/T_s) = -20 \text{ db}$$

$$(EIRP) = P_t * G_t$$

$$P_t = 2 \text{ Watts}$$

$$G_t = 1$$

$$(EIRP) = 3 \text{ db}$$

$$(SNR)_1 = 3 - 20 - 142 + 228.6 = 70 \text{ db}$$

$$(SNR)_2 = 3 - 20 - 152 + 228.6 = 60 \text{ db}$$

$$\text{Maximum data rate} = (MDR) = (SNR) - (E_b/N_o)$$

where

(E_b/N_o) = Min ratio of signal power per bit to noise power density

= 15 db (typical value corresponding to Bit Error Rate of $10E-6$)

$$(MDR)_1 = 70 - 15 = 55 \text{ db} = 300 \text{ kbps}$$

$$(MDR)_2 = 60 - 15 = 45 \text{ db} = 30 \text{ kbps}$$

Uplink calculations:

$$P_t = 2 \text{ Watts}$$

$$G_t = 10 \text{ db}$$

$$(EIRP) = 13 \text{ db}$$

$$G_r = 0 \text{ db}$$

$$T_s = 1000 \text{ K}$$

$$(G_r/T_s) = -30 \text{ db}$$

$$(FSL)_1 = 142 \text{ db}$$

$$(FSL)_2 = 152 \text{ db}$$

$$(SNR)_1 = 13 - 30 - 142 + 228.6 = 70 \text{ db}$$

$$(SNR)_2 = 13 - 30 - 152 + 228.6 = 60 \text{ db}$$

8.2.5 Maximum Doppler Shift

(to be determined)

8.2.6 Contact Time Calculations

Radius of earth	6378.165 km
Height of satellite	550 km
Value of u	3.986032 10E5
Orbital Period	95.65 min
Ground antenna min elevation angle	15 degrees
Coverage Sector angle at ground	75+75 degrees
Coverage Sector angle at earths center	37 deg
Coverage Time	10 min

8.2.7 System Reliability/Redundancy Requirements

Since this is our main communication link with the satellite a high level of accuracy and system reliability is required. Typically a bit error of one in 1000000 is considered reasonable. To achieve high degree of reliability following

8.2.8 System Weight/Size

(to be determined)

8.2.9 Antenna Design

Ground Tx/Rx Antenna:

A directional tracking antenna would be employed for Ground Station in order to achieve maximum contact time per pass and reasonably high directivity.

Satellite Tx/Rx Antenna:

(to be determined)

8.2.10 Power Consumption

On-board Transmitter	6 watts (only during transmission)
On-board Receiver	2 watts (operational mode)
	1 watt (standby mode)

8.2.11 System Cost

(to be determined)

8.2.12 Manufacturer Information

1. Cynetics Corporation
P.O. 2422
3824 Jet Drive
Rapid City , SD 57709
Phone (605) 394-6430
Fax (605) 394-6456
2. Motorola Inc.
Strategic Electronics Division
Advanced Programs Development
2501 S. Price Road
Chandler , AZ 85248-2899
Phone (602)732-3015

8.3 COMM-2

8.3.1 Purpose

This experimental communication system aims at establishing a data link with the satellite from a small inexpensive and portable ground transponder. The idea is to have multiple ground transponders (from 100 to 200) spread all over the state which could relay some kind of local data (like temperature, pressure, humidity, wind velocity etc.) up to the satellite whenever the satellite passes over them .

8.3.2 Data Rate / Bandwidth

Average contact time per pass	8 min or 480 sec.
Average number of passes per day	4
Data uplink per transponder per pass	24k bits
200 transponders	4.8M bps

System data rate	4.8M/480 or 10k bps
System bandwidth	12k Hz

8.3.3 Frequency Selection

The recent World Administrative Radio Conference (WARC) in Spain allocated new frequencies for small satellite systems . These are:

<u>Frequency Band</u>	<u>Purpose</u>
137.000---137.025 MHz	Primary
137.175---137.825 MHz	
148.000---149.900 MHz	
400.150---401.000 MHz	
137.025---137.175 MHz	Secondary
137.825---138.000 MHz	

Although the ultimate selection of frequency is subject to the availability of a particular frequency band and approval by the FCC, following are the relative merits and demerits of above mentioned frequency bands:

- At higher frequencies the available bandwidth is larger compared to lower frequencies .
- At higher frequencies antenna sizes become smaller , thus easier to manage .
- Free space loss is higher at higher frequencies .
- Doppler shift is more significant at higher frequencies than lower ones .

Considering all these factors and our system requirement it appears that the choice of higher frequency band would be more suitable for our satellite .

8.3.4 Multiple Access Technique

Various possible multiple access techniques were studied and the relative merits and demerits were evaluated.

- Frequency Division Multiple Access:

1. Each of the 200 Ground Transponders will have a distinct carrier frequency. Data will be uplinked at a rate less than 100bps using BPSK. Inter-channel spacing will be extremely narrow, ie. of the order of few hundred hertz.

2. The receiver on-board satellite (Rx-2) will receive all the 200 channels, downconvert them to an IF frequency and then separate them out. Data of each channel can be stored in the memory or directly sent to the downlink transmitter (Tx-2). Channel separation for so close channels could be very complex.

3. Each Ground Transponder will have a small very simple receiver that would detect the presence of signal coming from the TT&C transmitter (Tx-1) on-board satellite.

4. When the detected signal crosses certain threshold, it would be an indication that the satellite is within the range of the Ground Transponder, so it will start transmitting the stored data and will continue transmission until the satellite gets out of range.

5. It is expected that each Ground Transponder will barely manage to uplink all the data that was required to be transferred.

b. FDMA(Bent-pipe scheme):

1. Each of the 200 Ground Transponders will have a distinct carrier frequency. Data will be uplinked at a rate less than 100bps using BPSK. Inter-channel spacing will be extremely narrow, ie. of the order of few hundred hertz.

2. The receiver on-board satellite will receive all the 200 channels together as a single band and will convert the whole block to an IF level. The IF signal will be fed to the downlink transmitter Tx-2 for re-transmission to the Ground-Station.

3. At the Ground-Station the receiver will split the individual channels apart and detect the data in each channel.

4. Each Ground Transponder will have a small very simple receiver that would detect the presence of signal coming from the TT&C transmitter (Tx-1) on-board satellite.

5. When the detected signal crosses certain threshold, it would be an indication that the satellite is within the range of the Ground Transponder, so it will start transmitting the stored data and will continue transmission until the satellite gets out of range.

c. Packet Data Transfer Scheme:

1. All the 200 Ground Transponder will have identical transmitter that share a common frequency band of around 50Khz.

2. Each Ground-Transponder will have a receiver to detect the presence of satellite and to decode various messages from the satellite.

3. When a Ground-Transponder detects the presence of satellite, it transmits its ID and a "data transfer request" message.

4. The satellite receiver Rx-2, receives and decodes the ID and the message and transmits a "request acknowledge" message along with the ID of the specific Ground-Transponder, using another transmitter Tx-3.

5. The Ground-Transponder receives the message and starts transmitting its data packet.

6. When the satellite receiver Rx-2, receives the end-of-data header, it transmits a "receipt acknowledge" message.

7. The "request acknowledge" signal from the satellite serves as an indication to the other Ground-Transponders that the channel is busy while the "receipt acknowledge" message indicates that the channel has become idle. So another Ground-Transponder can "ask" for data transfer.

8. The only possibility of Ground-Transponder data clash is during the initial data transfer request phase. If that happens, the satellite receiver won't be able to decode the message so the satellite will transmit a "data clash" message.

9. In such a situation, the Ground-Transponders will retry the "data transfer request" after some random delay to avoid a clash again.

10. The transmission of various messages from the satellite to the Ground-Transponders will be quite frequent, ie. for each Ground-Transponder to transfer data, at least two messages needs to be transmitted

by the satellite. This amounts to an average of about 400 messages per pass of about 8 minutes.

Comparison of Various Schemes:

1. FDMA Scheme:

Advantages:

1. No additional transmitter on-board satellite is required.
2. Ground-Transponder receiver can be very simple.

Disadvantages:

1. Satellite receiver design for the separation of closely spaced channels can be very complex.
2. Ground-Transponder transmitter design would be tough, since the demand on frequency stability would be less than 100 hz at around 200 Mhz.
3. Doppler frequency shift can be a serious problem.

2. Bent-Pipe Scheme:

Advantages:

1. No additional transmitter on-board satellite is required.
2. Ground-Transponder receiver can be very simple.
3. No on-board data storage is required.
4. Satellite receiver would be very simple.

Disadvantages:

1. Ground-Transponder Transmitter design would be tough, since the demand on frequency stability would be less than 100 hz at around 200 Mhz.
2. Doppler frequency shift can be a serious problem.
3. Ground-Transponder needs to transmit higher power to maintain the same bit-error-rate.

3. Packet Data Transfer Scheme:

Advantages:

1. Simple satellite receiver design.
2. Much simpler Ground-Transponder transmitter design.
3. Doppler frequency shift won't be a serious problem.

Disadvantages:

1. Additional transmitter on-board satellite (Tx-3) is required.
2. Ground-Transponder needs a relatively complex receiver to decode various messages from the satellite.

8.3.5 Link Calculations

Following link equation is used for link calculations:

$$(SNR) = (EIRP) + (Gr/Ts) - (FSL) + (228.6)$$

where

(SNR)	= signal to noise ratio
(EIRP)	= equivalent isotropic radiated power
(Gr)	= receiver antenna gain
(Ts)	= equivalent receiver noise temperature
(FSL)	= free space loss

Link calculations are done at two different frequencies .

$$f_1 = 150 \text{ MHz wavelength}_1 = w_1 = 2.0 \text{ m}$$

$$f_2 = 450 \text{ MHz wavelength}_2 = w_2 = .67 \text{ m}$$

Minimum antenna angle above horizon = 30 degrees

Orbital altitude = 550 km

Maximum distance between satellite and ground station

$$R = 550 / \sin 30 \text{ or } 1100 \text{ km}$$

$$(FSL) = (4\pi R / \lambda)^2$$

$$(FSL)_1 = 137 \text{ db}$$

$$(FSL)_2 = 146 \text{ db}$$

Uplink calculations:

Pt	2 Watts
Gt	10 db
(EIRP)	13 db
Gr	0 db
Ts	1000 K
(Gr/Ts)	-30 db
(FSL)1	137 db
(FSL)2	146 db

$$\text{Maximum data rate} = (\text{MDR}) = (\text{SNR}) - (\text{Eb/No})$$

where

$$(\text{Eb/No}) = \text{Min ratio of signal power per bit to noise power density}$$

$$= 15 \text{ db (typical value corresponding to Bit Error Rate of } 10\text{E-6)}$$

$$(\text{MDR})1 = 70 - 15 = 55 \text{ db} = 300\text{k bps}$$

$$(\text{MDR})2 = 60 - 15 = 45 \text{ db} = 30\text{k bps}$$

$$(\text{SNR})1 = 13 - 30 - 142 + 228.6 = 70 \text{ db}$$

$$(\text{SNR})2 = 13 - 30 - 152 + 228.6 = 60 \text{ db}$$

8.3.6 Maximum Doppler Shift

(to be determined)

8.3.7 Contact Time Calculations

Radius of earth	6378.165 km
Height of satellite	550 km
Value of u	3.986032 10E5
Orbital Period	95.65 min
Ground antenna min elevation angle	30 deg degrees
Coverage Sector angle at ground	60+60 deg
Coverage Sector angle at earths center	30 deg
Coverage Time	8 min

8.3.8 System Reliability / Redundancy

Since this would be an experimental system its reliability constraints both in terms of bit-error-rate and equipment redundancy would be much lesser than COMM-1 . A BER of $10E-5$ is considered suitable for the type of data to be normally handled by this system .

8.3.9 System Weight / Size

(to be determined)

8.3.10 Antenna Design

(For Satellite Receiver)

It is assumed that the frequency bands for all the four links of COMM-1 and COMM-2 (ie. up and down links of COMM-1 and up and down links of COMM-2) are adjacent to each other. Therefore a common antenna can be used for both COMM-1 and COMM-2. A circulator would be required to connect the onboard Transmitter(s) and Receiver(s) to a common antenna. The uplink signals of COMM-1 and COMM-2 would be separated by means of suitable filters, while the downlink signals of COMM-1 and COMM-2 can be combined through a matching unit .

8.3.11 Power Consumption

On-board Receiver	
operational mode	4 watts
standby mode	2 watts

8.3.12 System Cost

(to be determined)

8.3.13 Perspective Suppliers

- a. Cynetics Corporation
P.O. Box 2422
3824 Rapid City , SD 37709

Appendix A

Figures and Tables

Appendix A.1

Integration

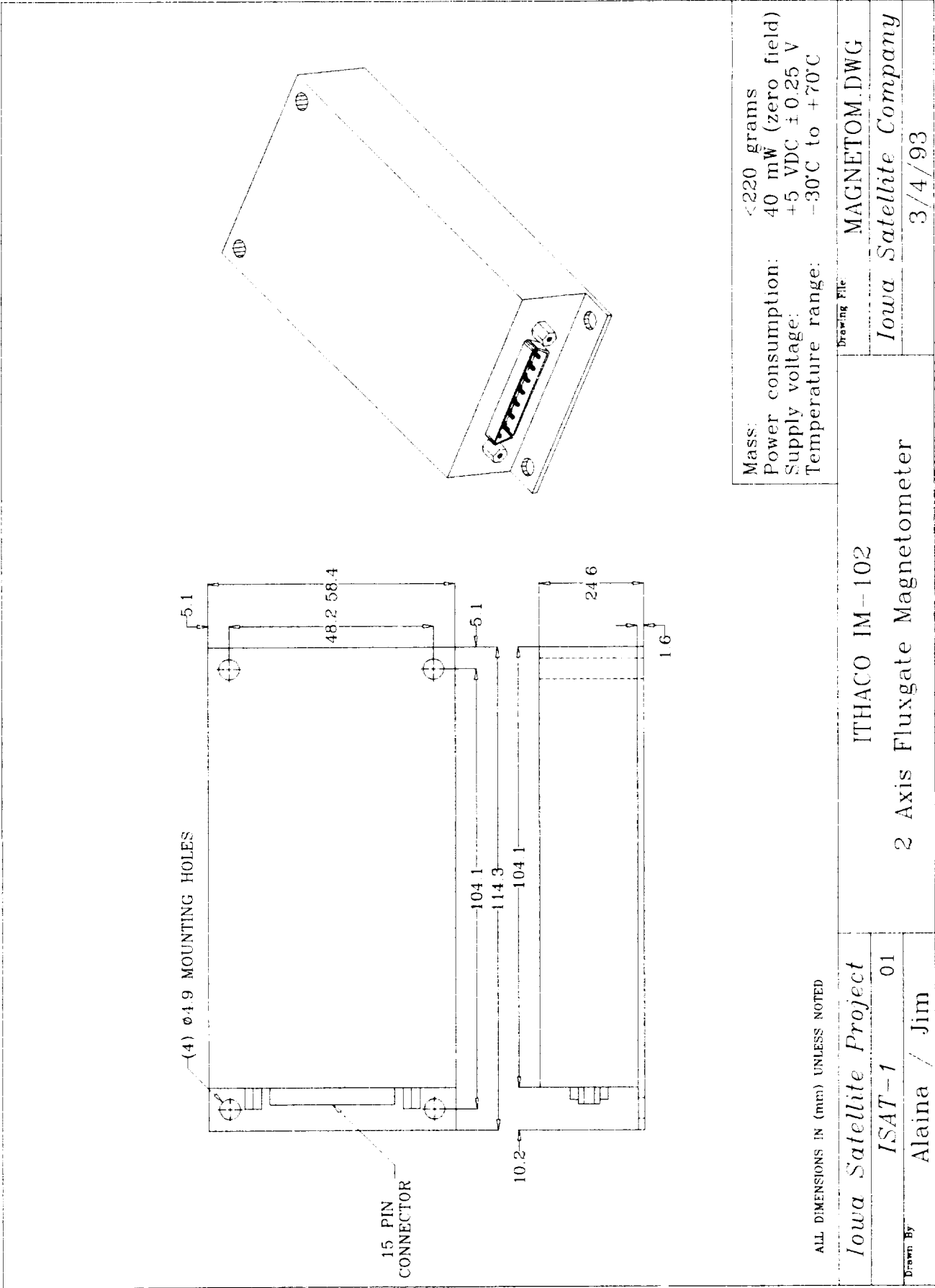


Figure A.1.1 Magnetometer

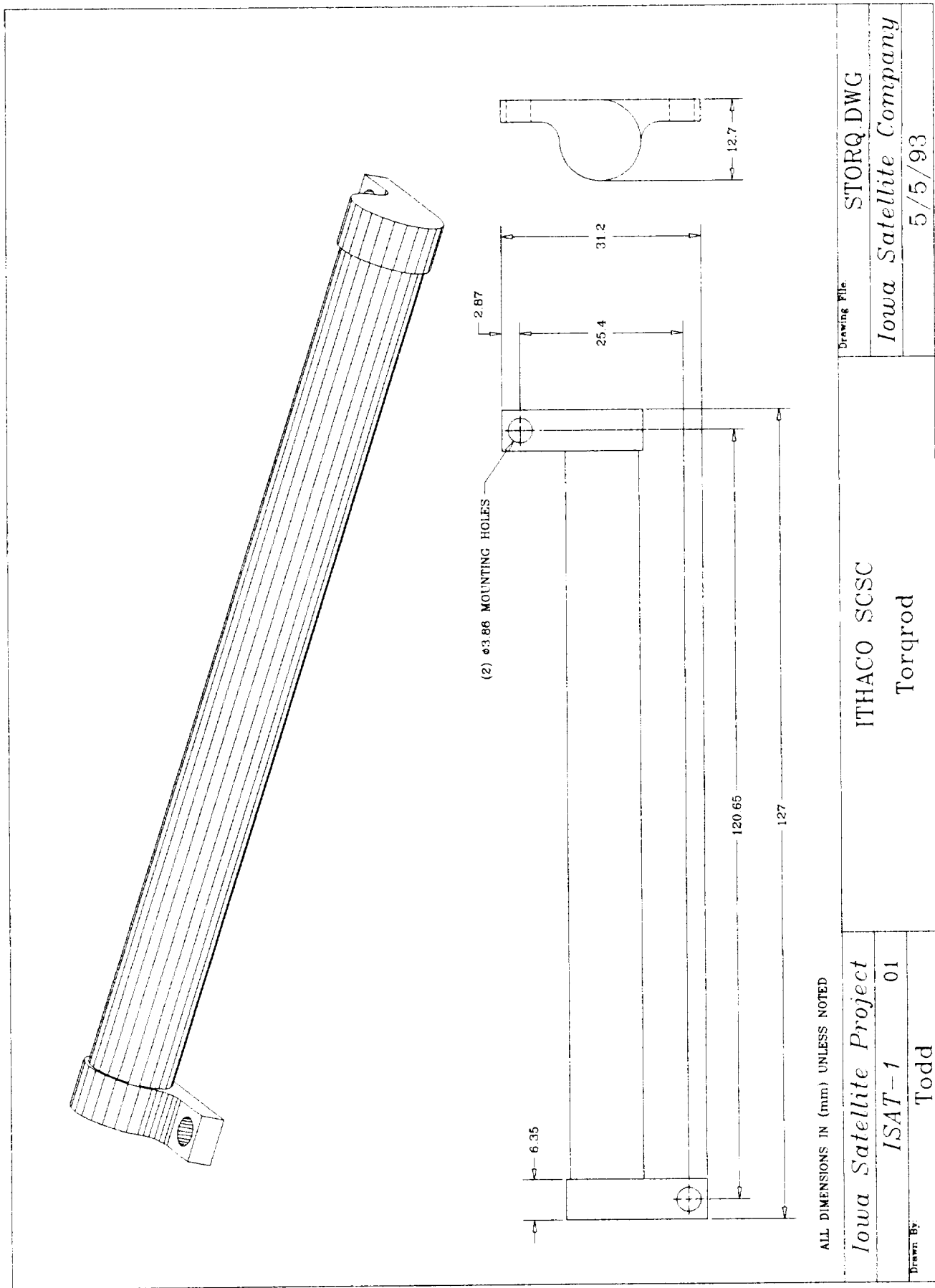
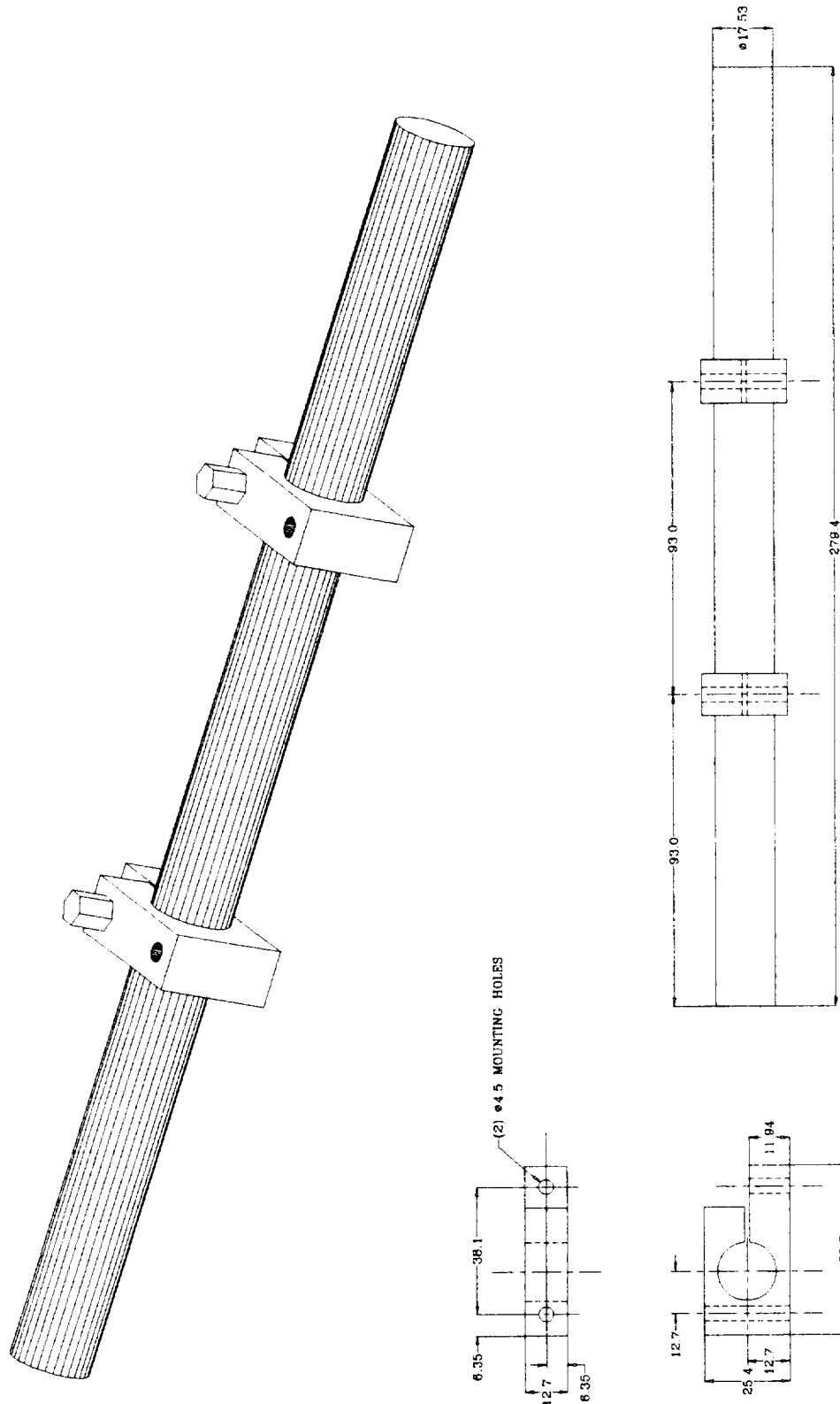


Figure A.1.2 Torqrod



ALL DIMENSIONS IN (mm) UNLESS NOTED

Iowa Satellite Project

ISAT-1 01

Alaina

ITHACO TR10CFR

Torqrod

Drawing File

LTORQ.DWG

Iowa Satellite Company

5/5/93

Figure A.1.3 Torqrod

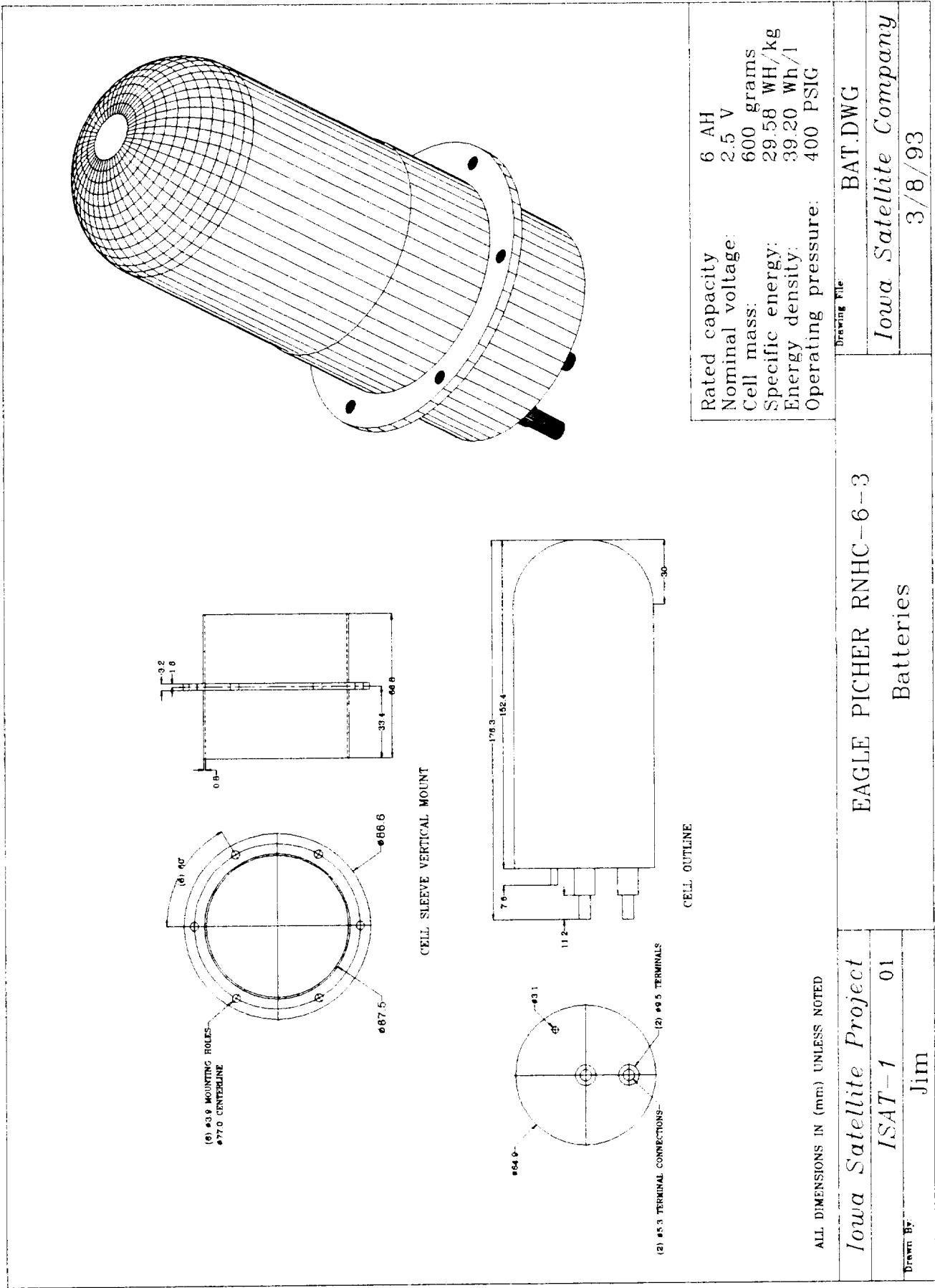


Figure A.1.4 Batteries

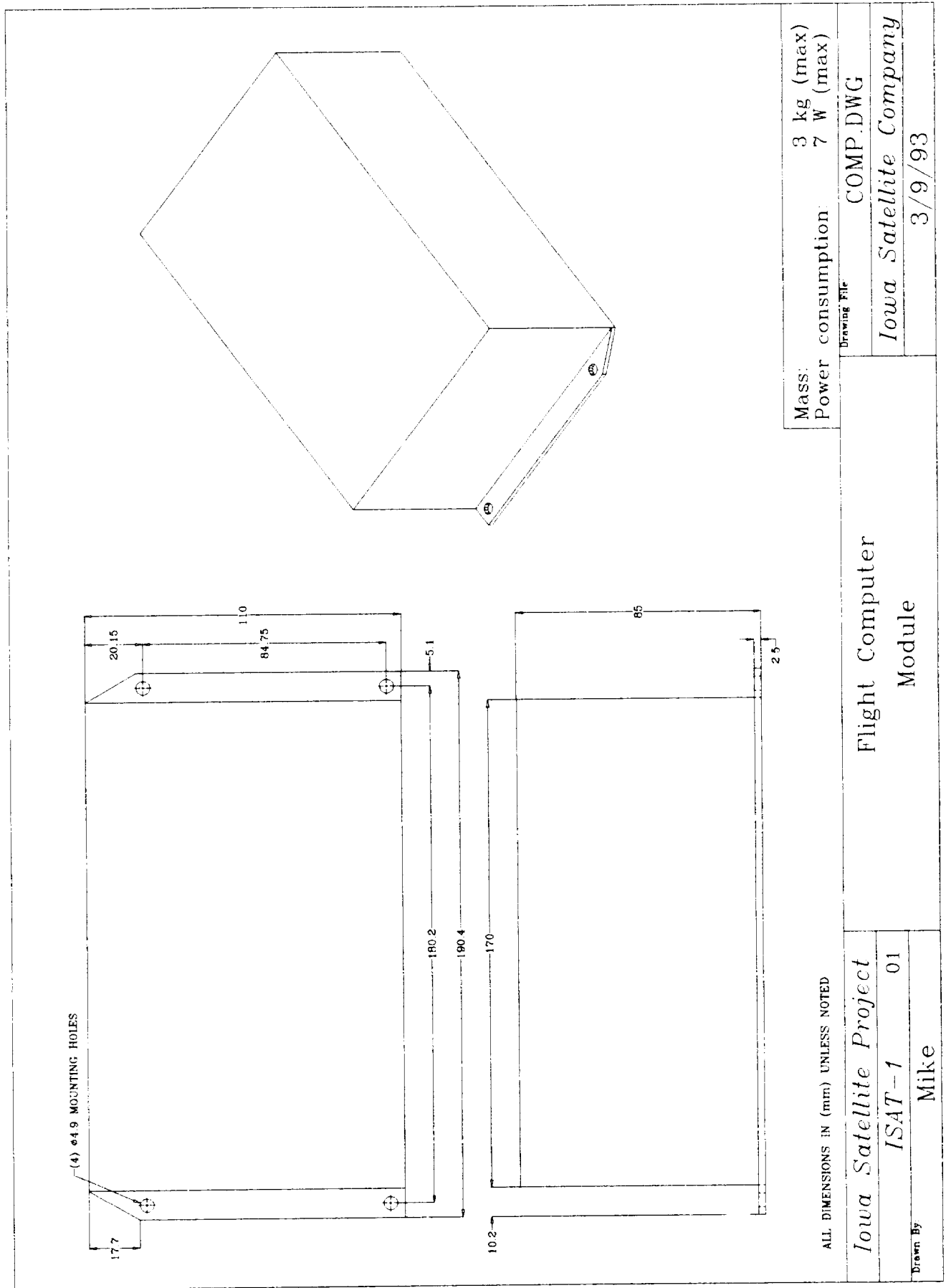


Figure A.1.5 Flight Computer

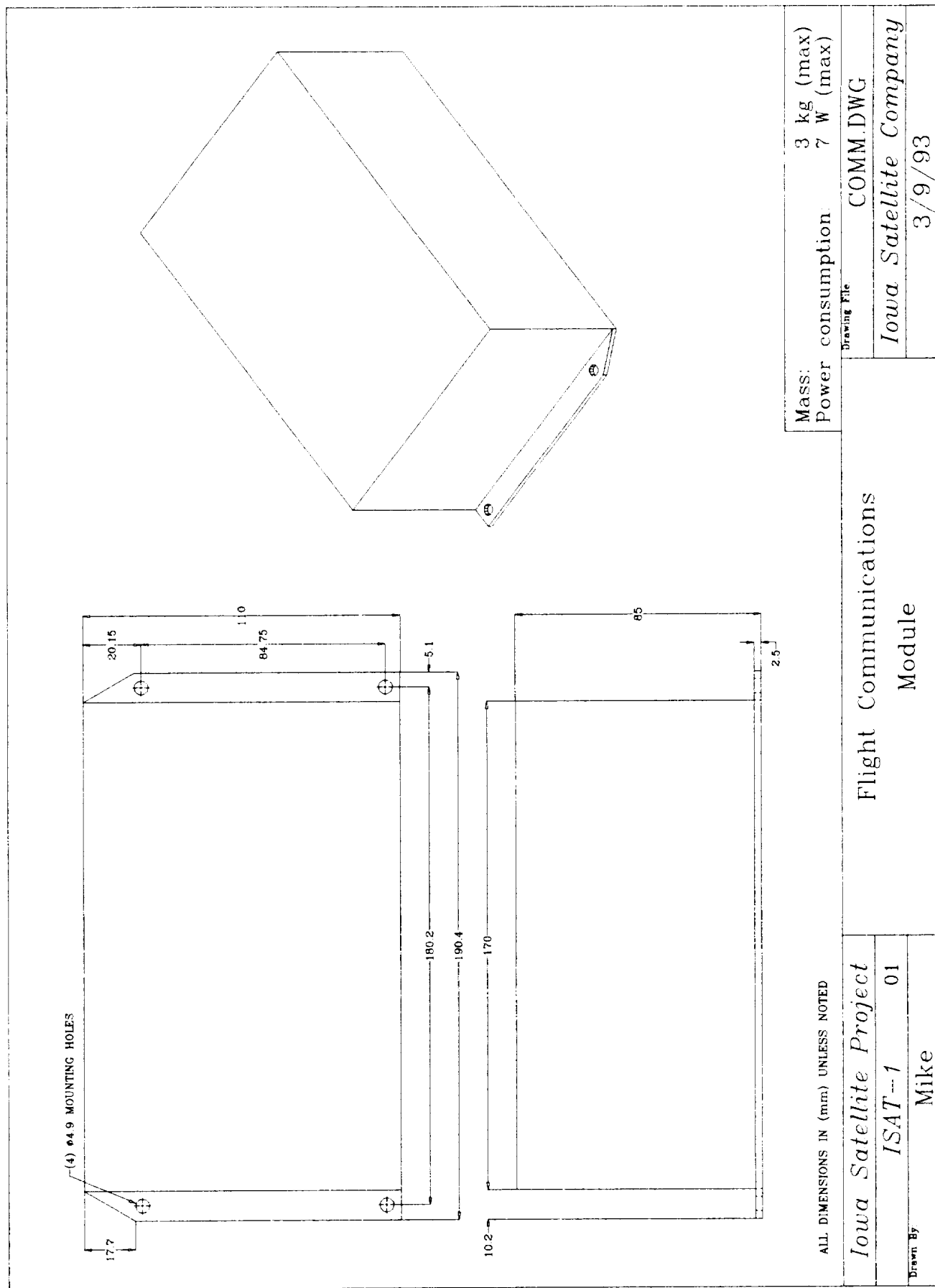


Figure A.1.6 Flight Communications

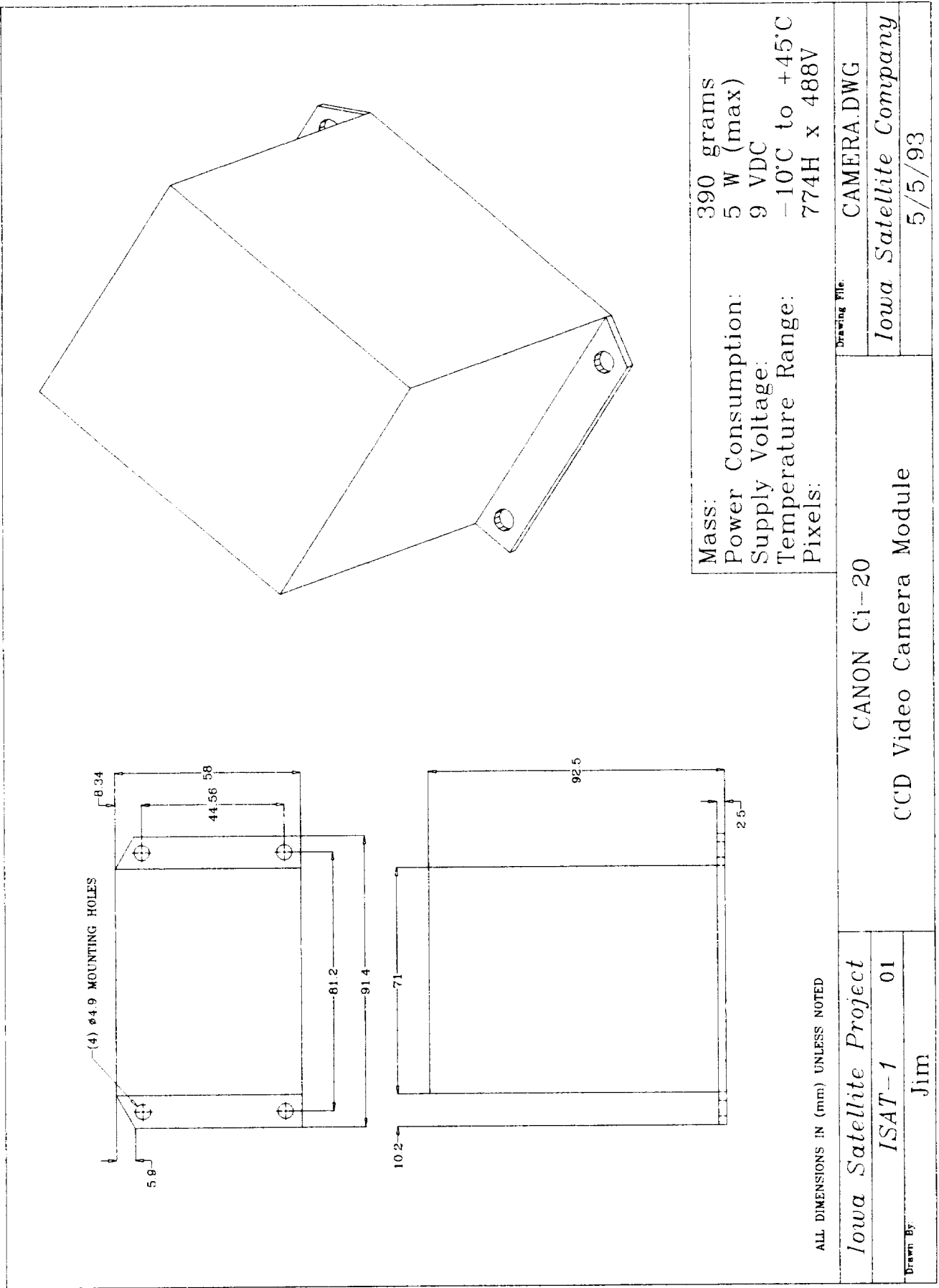


Figure A.1.7 Camera Module

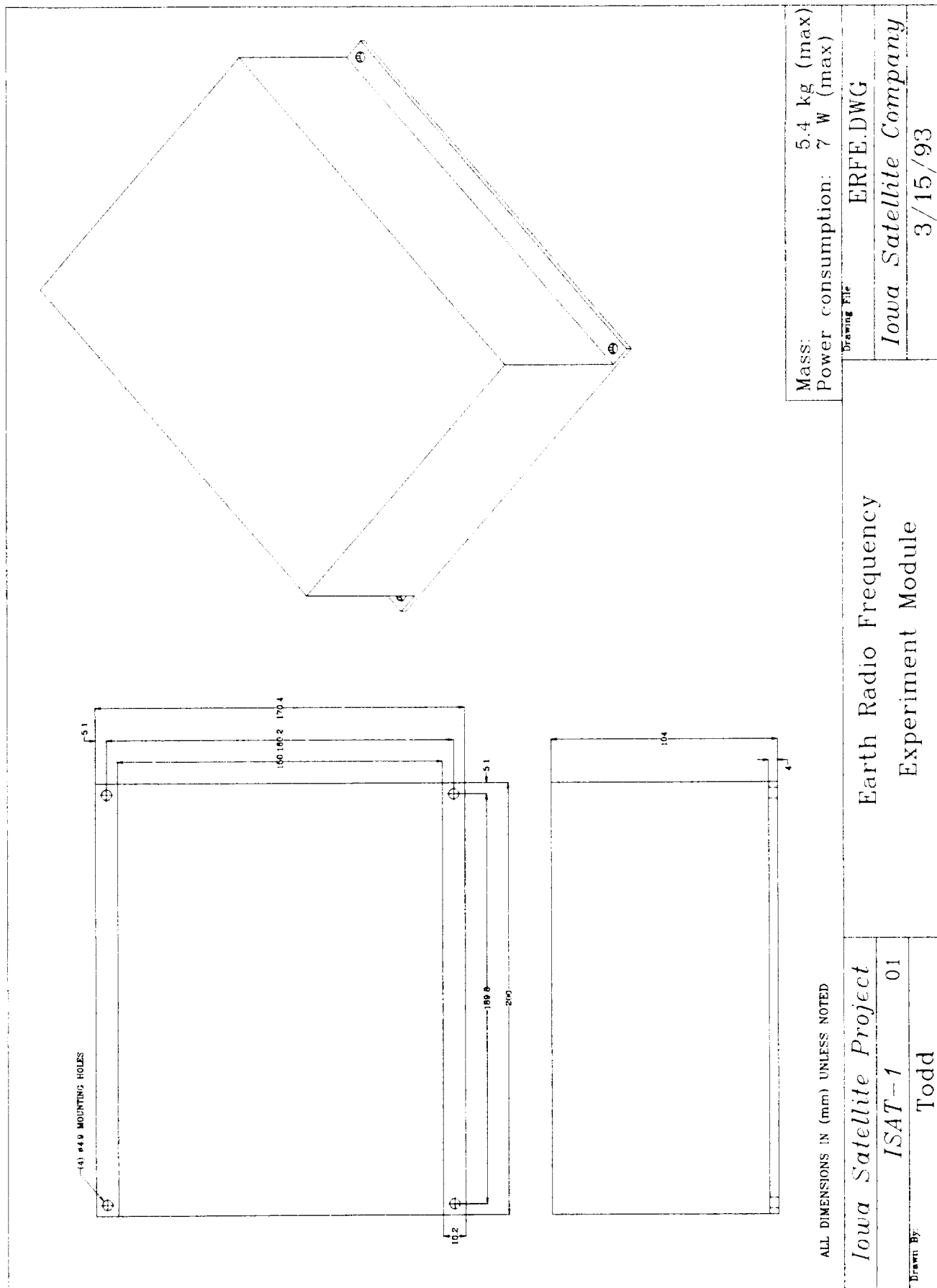


Figure A.1.8 E.R.F.E. Module

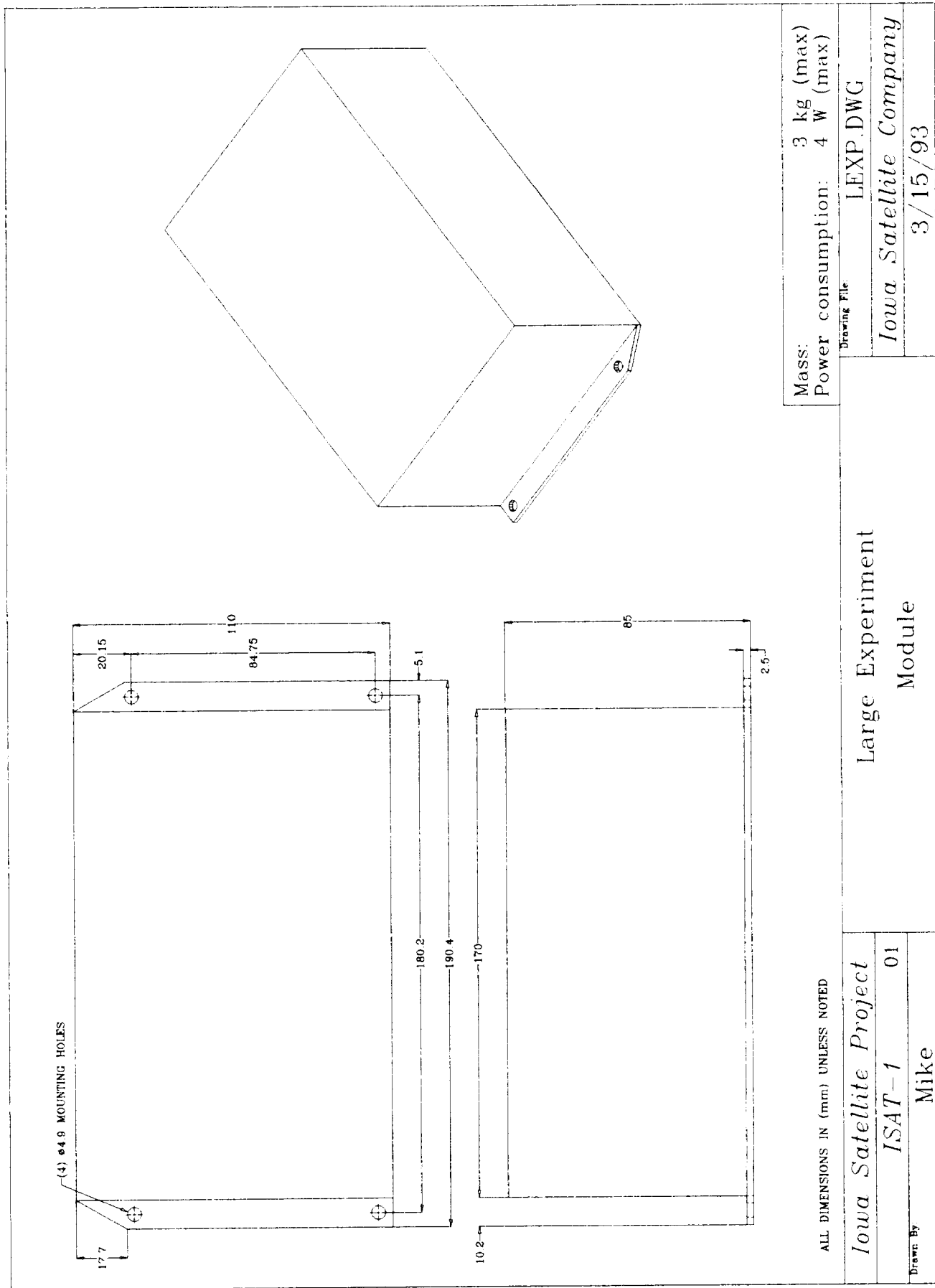


Figure A.1.9 Experiment Module

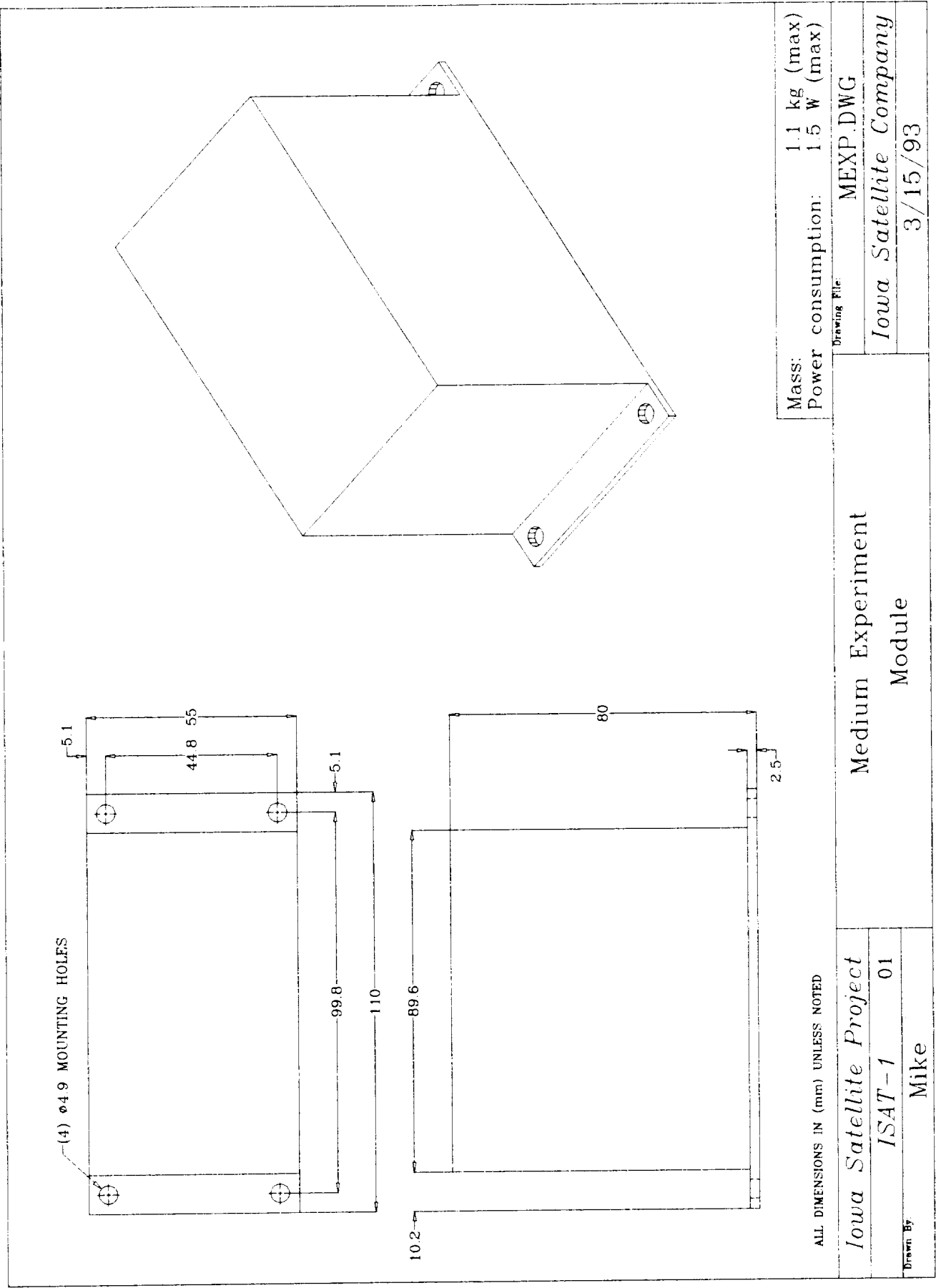


Figure A.1.10 Experiment Module

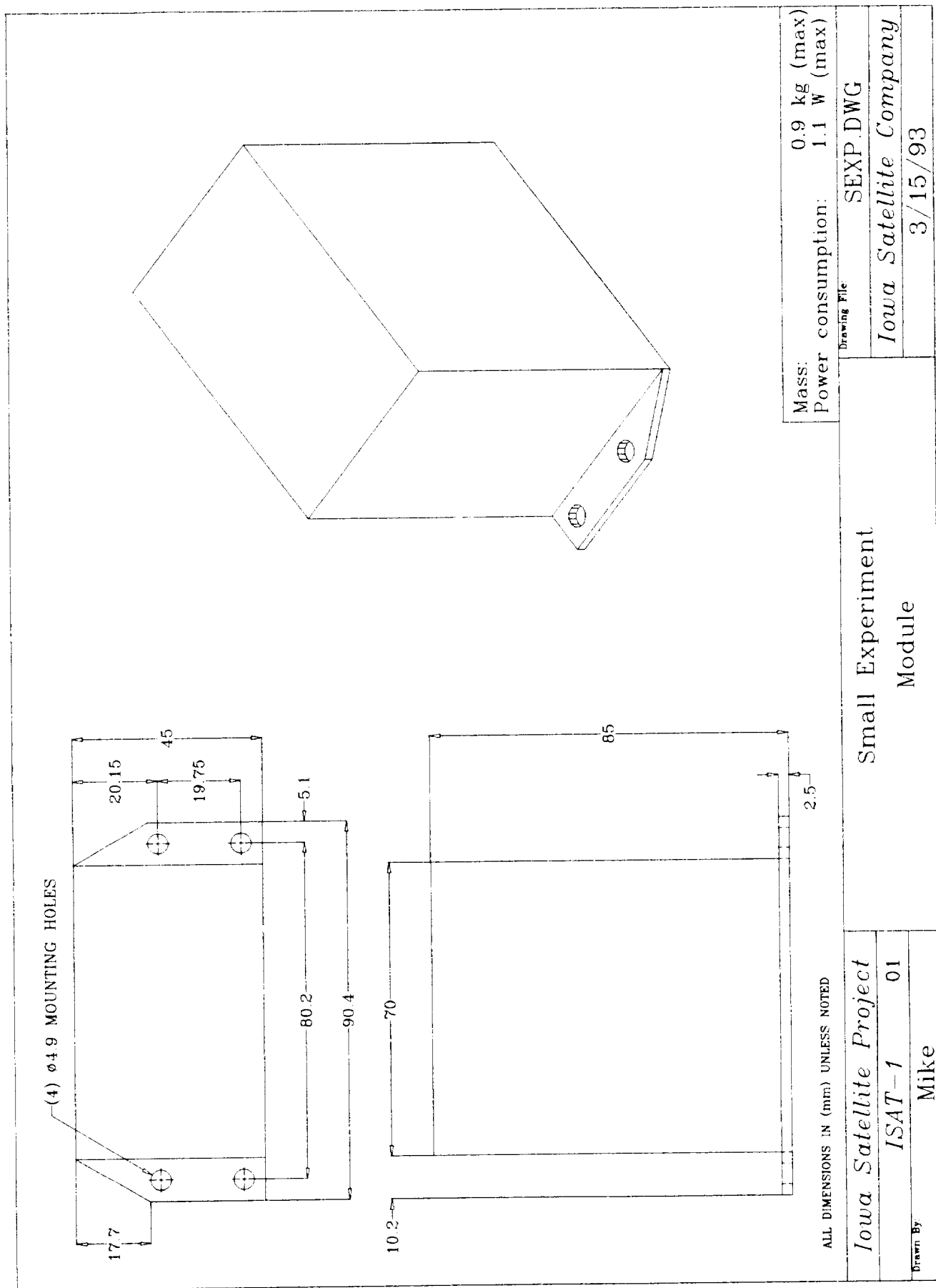


Figure A.1.11 Experiment Module

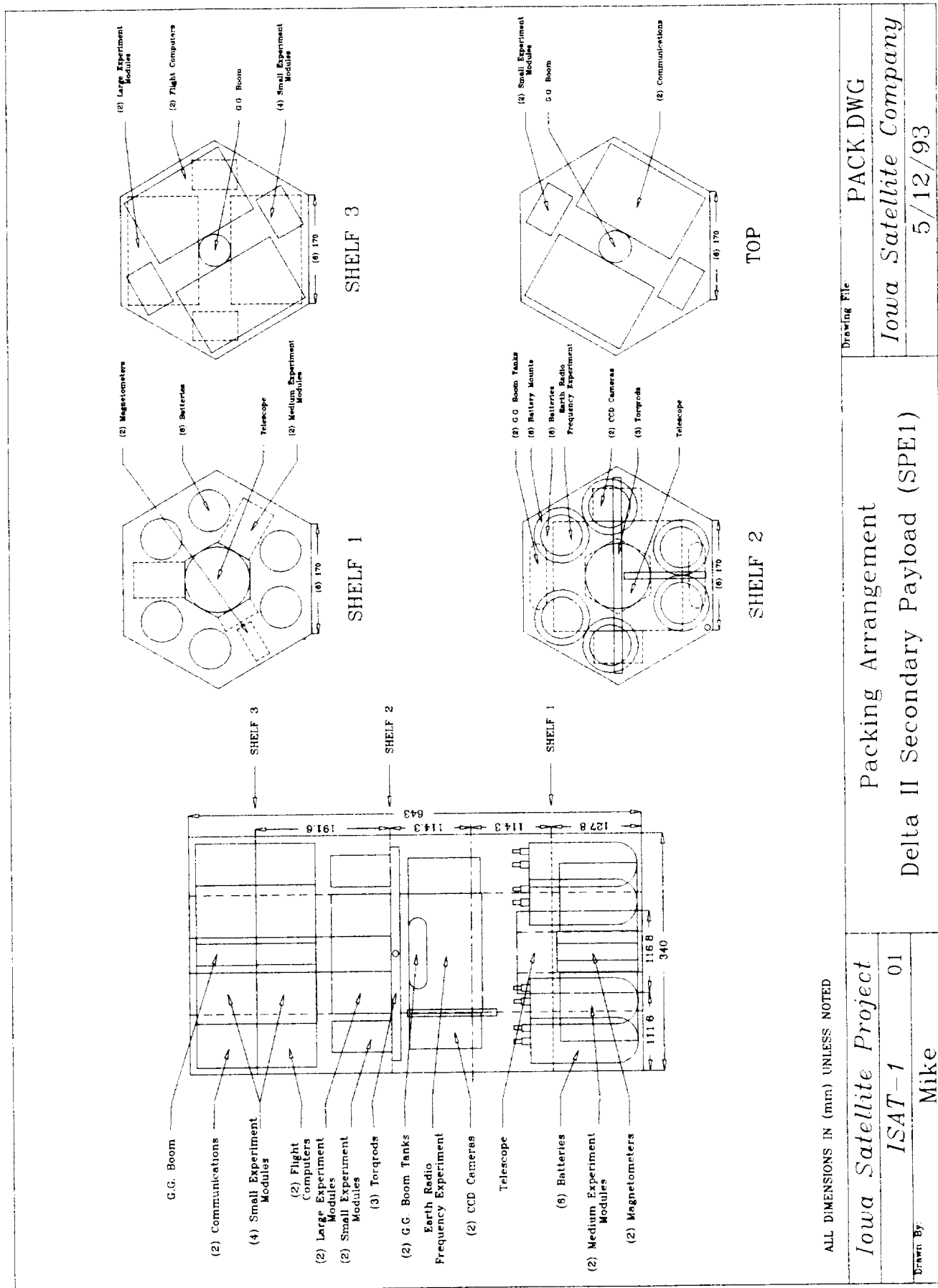


Figure A.1.12 Packing Arrangement

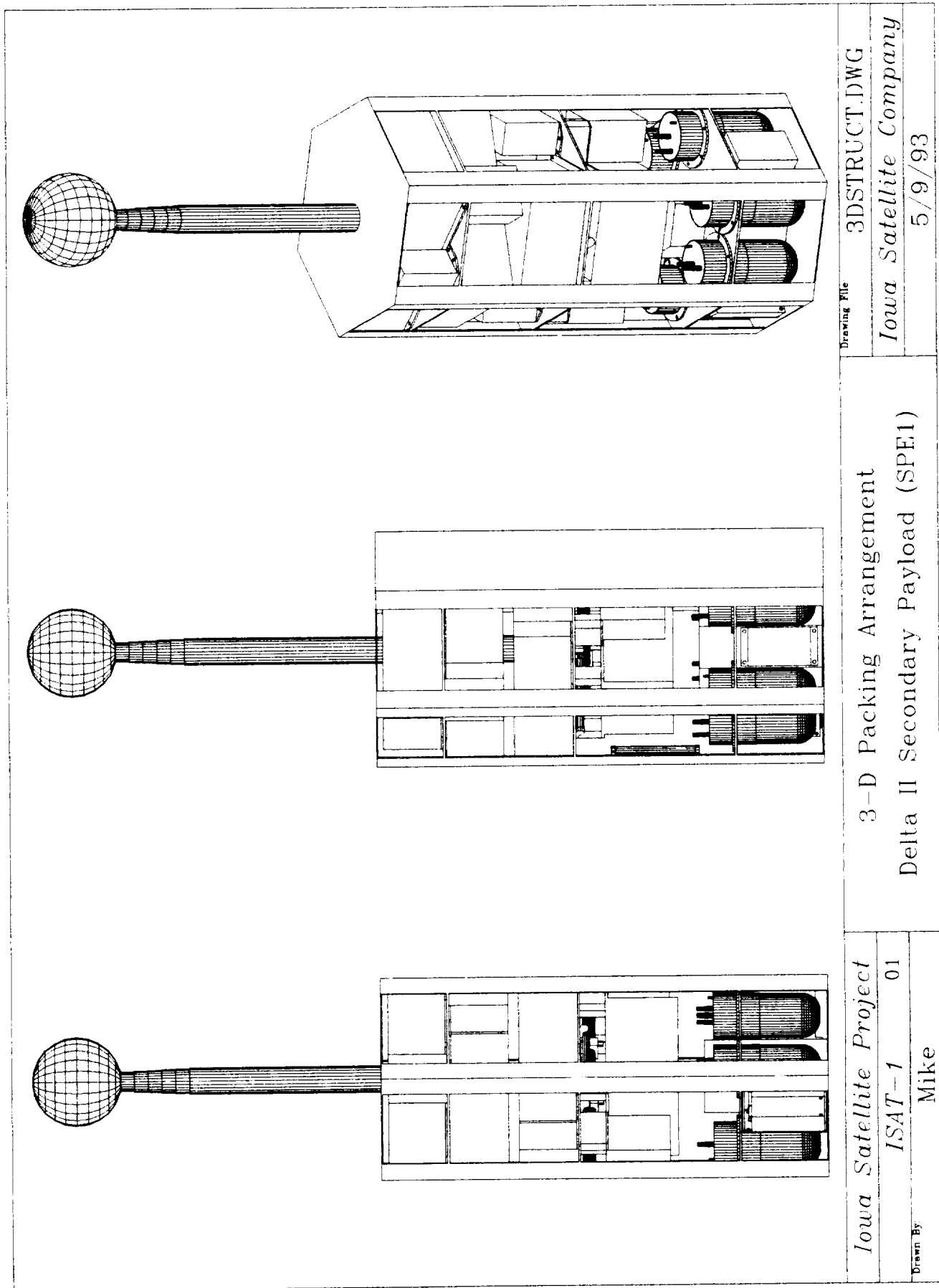


Figure A.1.13 Packing Arrangement

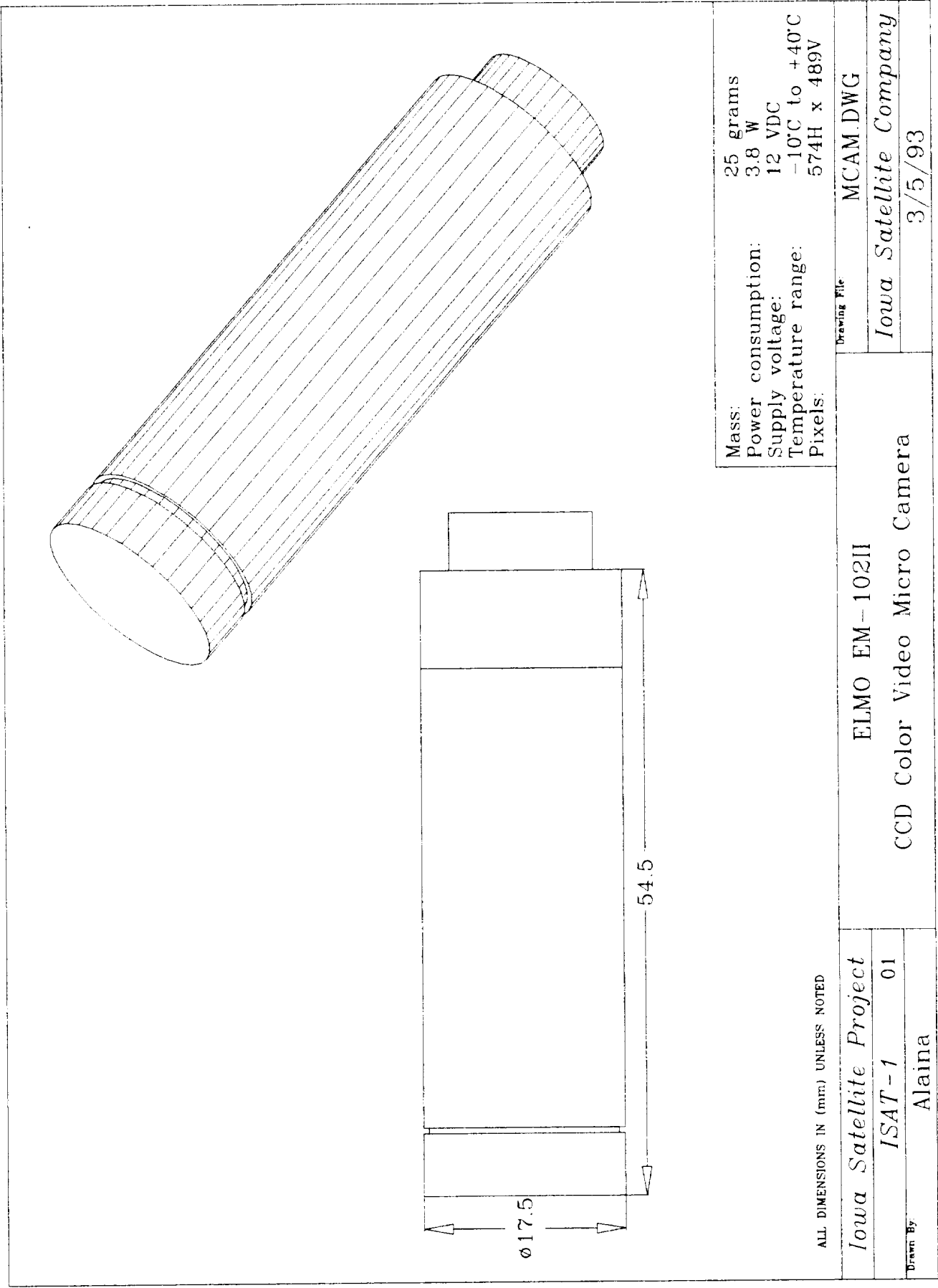


Figure A.1.14 Micro Camera

Appendix A.2

Structures

Iowa Satellite Project
Spacecraft Bus/Payload

#	Component	Mass (kg)	Power (W)	Dimensions
1	Structure	10	0	T.B.D.
1	G.G. Boom	7	0	50.8 mm diameter (max)
2	Communications	3	7	170x110x85 mm
2	Flight Computer	3	7	170x110x85 mm
2	Large Experiment Module	3	4	170x110x85 mm
2	Medium Experiment Module	1.1	1.5	110x55x80 mm
6	Small Experiment Module	0.9	1.1	90.4x45x85 mm
1	Pitch Torqrod	0.1	0.25	127x12.7 mm diameter
1	Roll Torqrod	0.1	0.25	127x12.7 mm diameter
1	Yaw Torqrod	0.45	2	381x25.4 mm diameter
2	G.G. Boom Tanks	0.1	0	25.4x101.6x101.6 mm (max)
1	Earth Radio Frequency Experiment	5.4	7	200x170.4x104 mm
2	CCD Camera	2	5	50.8x76.2x101.6 mm (max)
1	Telescope	3	0	101.6 mm diameter, 177.8 mm height (max)
6	Battery	0.6	0	64.9 mm diameter cylinder, 176.3 mm height with terminals
6	Battery Mount	0	0	86.6 mm diameter
2	Magnetometer	0.22	0.04	114.3x58.4x24.6 mm

Total Power (W)

65.18

Total Mass (kg)

59.89

Table A.2.1 Mass Budget

UNIV VERSION
MAY 3 1993
15:31:17
PLOT NO. 1
PREP7 ELEMENTS
TYPE NUM
BC SYMBOLS

YV =100
ZV =10
DIST=0.368007
ZF =0.3215

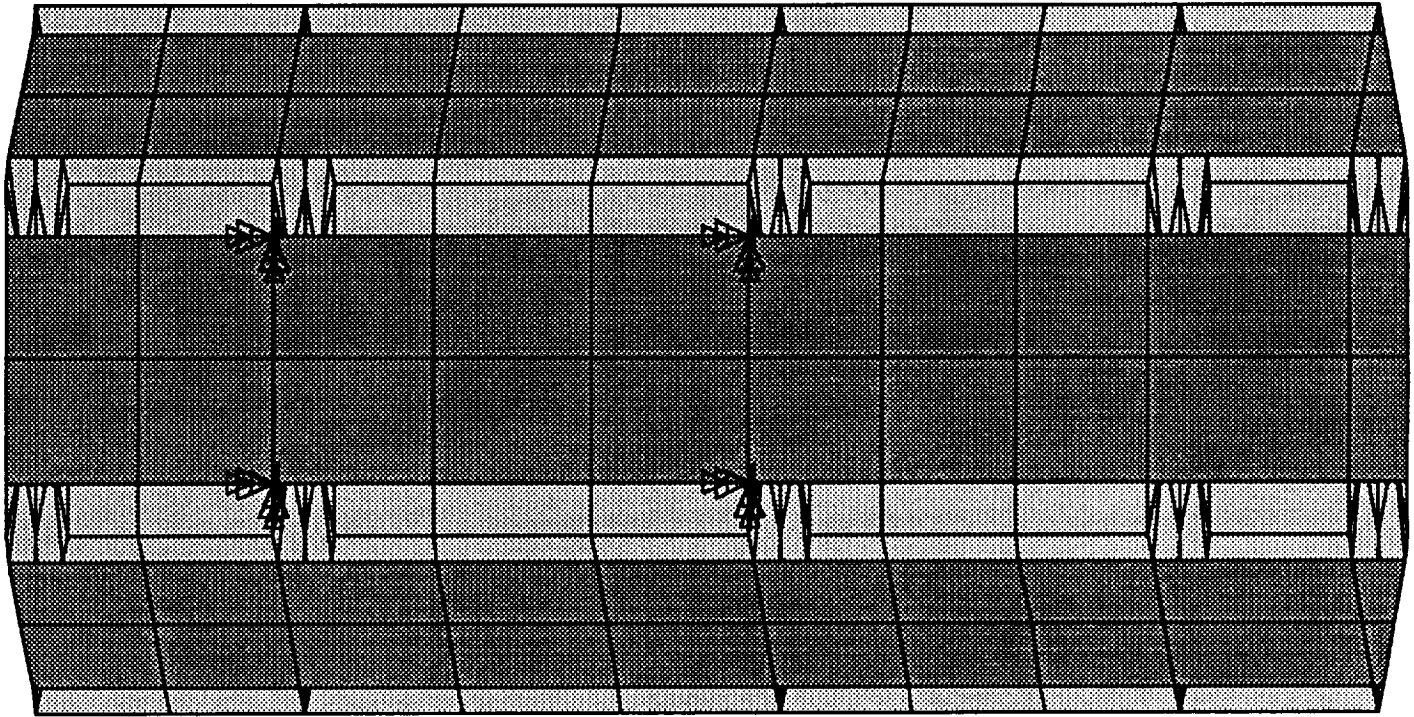


Figure A.2.1 Element Plot

ISAT Structural Mass Estimate						

note:	- same structural configuration for Al & Gr/Ep					
	- fasteners not included					
	- bus sensors/wiring/etc. not included					
	- Aluminum density taken from 6061-T6: 2710 kg/m^3					
	- Graphite Epoxy density: 1600 kg/m^3					

SIDE PANEL:		each panel is 0.170 by 0.643				
		area:	0.10931			

TOP/BOTTOM PANEL:		area of hexagon: (1/2)*apothem*perimeter				
		apothem:	0.17			
		perimeter	1.02			
		area	0.0867			

STRINGER:		a = outer flange length				
		t = thickness of stringer				
		area: t*a - 0.5(t**2)*(tan 30)				

SHELVES:		each shelf is a hexagon				
		area:	0.0867			

SIDE PANELS:						
	Panel	Total		Mass	Mass	Panel
	Thickness	Area	Volume	Al	Gr/Ep	Thickness
	(m)	(m^2)	(m^3)	(kg)	(kg)	(in)

	0.001016	0.65586	0.000666	1.80582	1.06617	0.040
	0.001366	0.65586	0.000896	2.42790	1.43345	0.054
	0.001716	0.65586	0.001125	3.04999	1.80073	0.068
	0.002066	0.65586	0.001355	3.67207	2.16801	0.081
	0.002416	0.65586	0.001585	4.29415	2.53529	0.095
	0.002766	0.65586	0.001814	4.91623	2.90257	0.109
	0.003116	0.65586	0.002044	5.53832	3.26986	0.123
	0.003466	0.65586	0.002273	6.16040	3.63714	0.136

TOP/BOTTOM PANEL:						
	Panel	Total		Mass	Mass	Panel
	Thickness	Area	Volume	Al	Gr/Ep	Thickness
	(m)	(m^2)	(m^3)	(kg)	(kg)	(in)

	0.001016	0.1734	0.00018	0.47743	0.28188	0.040
	0.001366	0.1734	0.00024	0.64190	0.37898	0.054

	0.001716	0.1734	0.00030	0.80637	0.47609	0.068	
	0.002066	0.1734	0.00036	0.97084	0.57319	0.081	
	0.002416	0.1734	0.00042	1.13531	0.67030	0.095	
	0.002766	0.1734	0.00048	1.29978	0.76740	0.109	
	0.003116	0.1734	0.00054	1.46425	0.86450	0.123	
	0.003466	0.1734	0.00060	1.62872	0.96161	0.136	
	0.003816	0.1734	0.00066	1.79319	1.05871	0.150	
	0.004166	0.1734	0.00072	1.95766	1.15582	0.164	
	0.004516	0.1734	0.00078	2.12213	1.25292	0.178	
	0.004866	0.1734	0.00084	2.28660	1.35002	0.192	
	0.005216	0.1734	0.00090	2.45107	1.44713	0.205	
	0.005566	0.1734	0.00097	2.61554	1.54423	0.219	
	0.005916	0.1734	0.00103	2.78001	1.64134	0.233	
	0.006266	0.1734	0.00109	2.94448	1.73844	0.247	
SHELVES:							
	Shelf	Total		Mass	Mass	Shelf	
	Thickness	Area	Volume	Al	Gr/Ep	Thickness	
	(m)	(m^2)	(m^3)	(kg)	(kg)	(in)	
	-----	-----	-----	-----	-----	-----	
	0.002032	0.2601	0.00053	1.43230	0.84564	0.080	
	0.002382	0.2601	0.00062	1.67900	0.99129	0.094	
	0.002732	0.2601	0.00071	1.92571	1.13695	0.108	
	0.003082	0.2601	0.00080	2.17241	1.28261	0.121	
	0.003432	0.2601	0.00089	2.41912	1.42826	0.135	
	0.003782	0.2601	0.00098	2.66582	1.57392	0.149	
	0.004132	0.2601	0.00107	2.91253	1.71957	0.163	
	0.004482	0.2601	0.00117	3.15923	1.86523	0.176	
	0.004832	0.2601	0.00126	3.40594	2.01089	0.190	
	0.005182	0.2601	0.00135	3.65264	2.15654	0.204	
	0.005532	0.2601	0.00144	3.89935	2.30220	0.218	
	0.005882	0.2601	0.00153	4.14605	2.44785	0.232	
	0.006232	0.2601	0.00162	4.39276	2.59351	0.245	
	0.006582	0.2601	0.00171	4.63946	2.73917	0.259	
	0.006932	0.2601	0.00180	4.88617	2.88482	0.273	
	0.007282	0.2601	0.00189	5.13287	3.03048	0.287	
	0.007632	0.2601	0.00199	5.37958	3.17613	0.300	
	0.007982	0.2601	0.00208	5.62628	3.32179	0.314	
	0.008332	0.2601	0.00217	5.87299	3.46745	0.328	
	0.008682	0.2601	0.00226	6.11969	3.61310	0.342	
	0.009032	0.2601	0.00235	6.36639	3.75876	0.356	
	0.009382	0.2601	0.00244	6.61310	3.90441	0.369	
	0.009732	0.2601	0.00253	6.85980	4.05007	0.383	
STRINGERS:							
	Flange	Flange	Total		Mass	Mass	Flange
	Length	Thickness	Area	Volume	Al	Gr/Ep	Thickness

	(m)	(m)	(m ²)	(m ³)	(kg)	(kg)	(in)
	-----	-----	-----	-----	-----	-----	-----
10%	0.0170	0.001016	1.697E-05	1.09E-05	0.01746	0.01746	0.040
	0.0170	0.001366	2.268E-05	1.46E-05	0.02334	0.02334	0.054
	0.0170	0.001716	2.832E-05	1.82E-05	0.02914	0.02914	0.068
	0.0170	0.002066	3.389E-05	2.18E-05	0.03487	0.03487	0.081
	0.0170	0.002416	3.939E-05	2.53E-05	0.04052	0.04052	0.095
	0.0170	0.002766	4.481E-05	2.88E-05	0.04610	0.04610	0.109
	0.0170	0.003116	5.017E-05	3.23E-05	0.05161	0.05161	0.123
15%	0.0255	0.001016	2.561E-05	1.65E-05	0.02635	0.02635	0.040
	0.0255	0.001366	3.429E-05	2.21E-05	0.03528	0.03528	0.054
	0.0255	0.001716	4.291E-05	2.76E-05	0.04414	0.04414	0.068
	0.0255	0.002066	5.145E-05	3.31E-05	0.05293	0.05293	0.081
	0.0255	0.002416	5.992E-05	3.85E-05	0.06165	0.06165	0.095
	0.0255	0.002766	6.832E-05	4.39E-05	0.07029	0.07029	0.109
	0.0255	0.003116	7.666E-05	4.93E-05	0.07886	0.07886	0.123
20%	0.0340	0.001016	3.425E-05	2.2E-05	0.03523	0.03523	0.040
	0.0340	0.001366	4.591E-05	2.95E-05	0.04723	0.04723	0.054
	0.0340	0.001716	5.749E-05	3.7E-05	0.05915	0.05915	0.068
	0.0340	0.002066	6.901E-05	4.44E-05	0.07100	0.07100	0.081
	0.0340	0.002416	8.046E-05	5.17E-05	0.08278	0.08278	0.095
	0.0340	0.002766	9.184E-05	5.91E-05	0.09448	0.09448	0.109
	0.0340	0.003116	0.0001031	6.63E-05	0.10611	0.10611	0.123

Table A.2.2 Structural Mass Estimate

Appendix A.3

Payload

REV LTR	DESCRIPTION	DATE
A	INITIAL RELEASE	3/10/91
B	END CAPS & PLUNGER	2/10/92

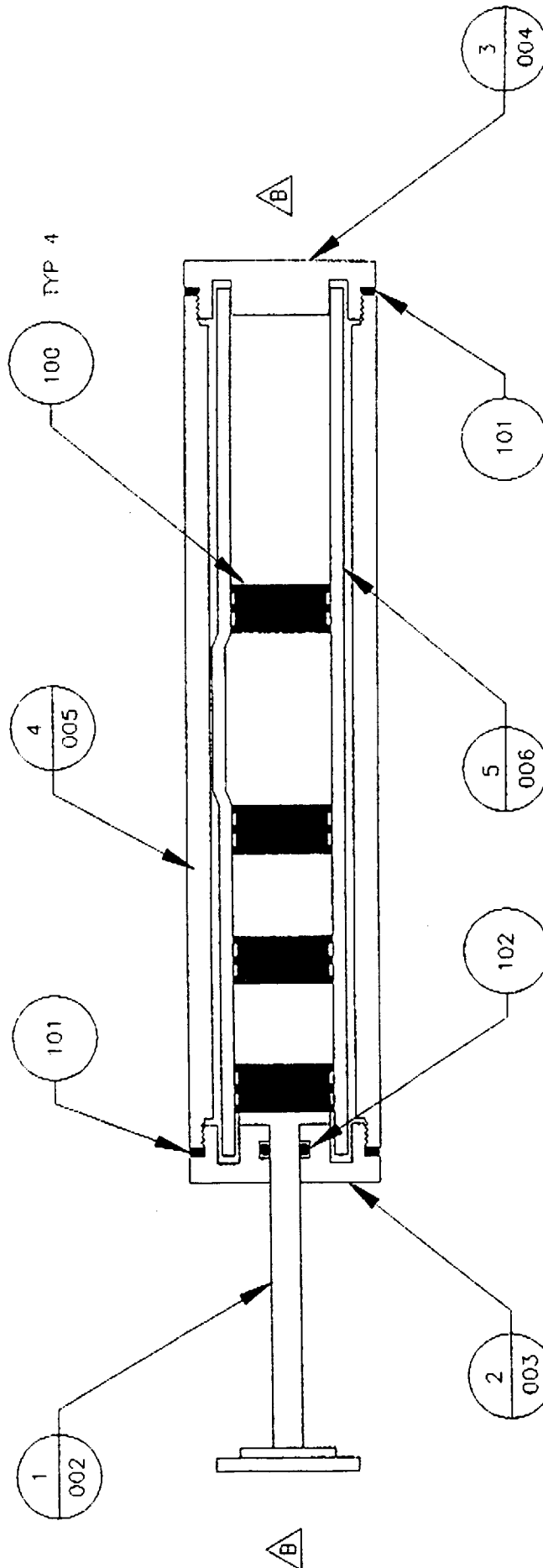


Figure A.3.1 Syringe Assembly

NEXT HIGHER ASSY:		BIOERVE SPACE TECHNOLOGIES CAMPUS BOX 429, UNIV. OF COLORADO @ BOULDER BOULDER, COLORADO 80309 PHONE: (303) 492-1005	
SCALE	NONE	APPROVED BY:	M. ROBINSON
DATE	3/10/91	PROJECT	USML-1
TITLE		REV	B
		SYRINGE ASSEMBLY	
DATE	SEE B.O.M.	SHT	OF 1
TOLERANCES: XX = ± 01, XXX = ± 005		DRAWING NUMBER: FPA-001	
FINISH:			

REV LTR	DESCRIPTION	DATE
A	INITIAL RELEASE	3/10/91

TUBING 1.0" DIA. X 1/8" WALL

7/8 X 32 UNF

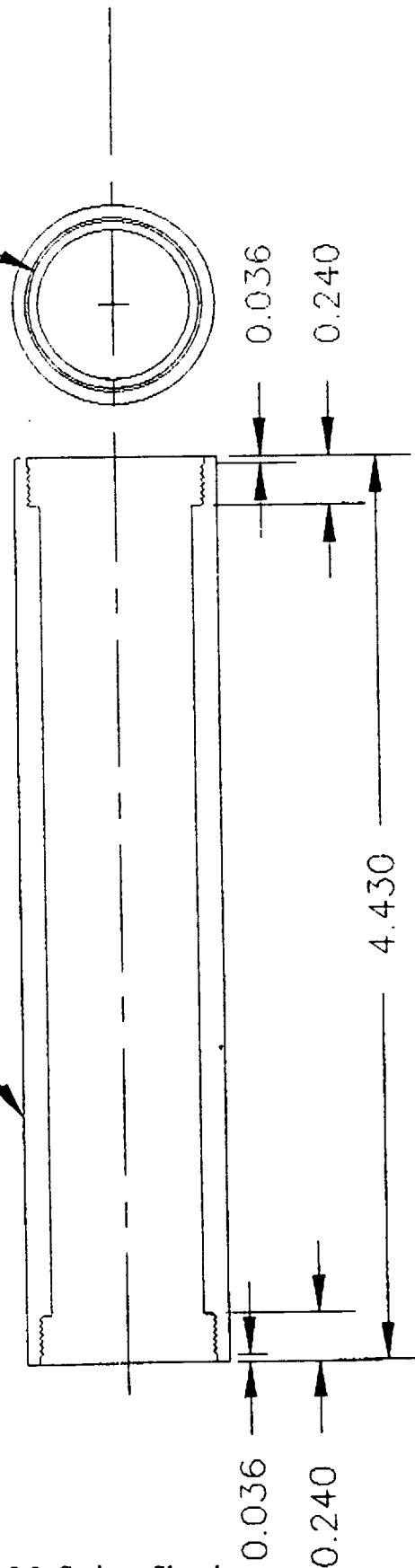


Figure A.3.2 Syringe Sheath

NEXT HIGHER ASSY:		BIOSERVE SPACE TECHNOLOGIES	
FPA-001		CAMPUS 90X 425, UNIV. OF COLORADO & BOULDER	
		BOULDER, COLORADO 80309 PHONE (303) 492-1005	
SCALE	NONE	APPROVED BY: M. ROBINSON	DRAWN BY: MCR
DATE	3/10/91	PROJECT: USML-1	REV: A
TITLE:		SYRINGE SHEATH	
HTL:	LEXAN	SH: 1	OF: 1
TOLERANCES: XX = ±.01, XXX = ±.005		DRAWING NUMBER:	
FINISH:		FPA-005	

Appendix A.7

Ground Systems and Operations

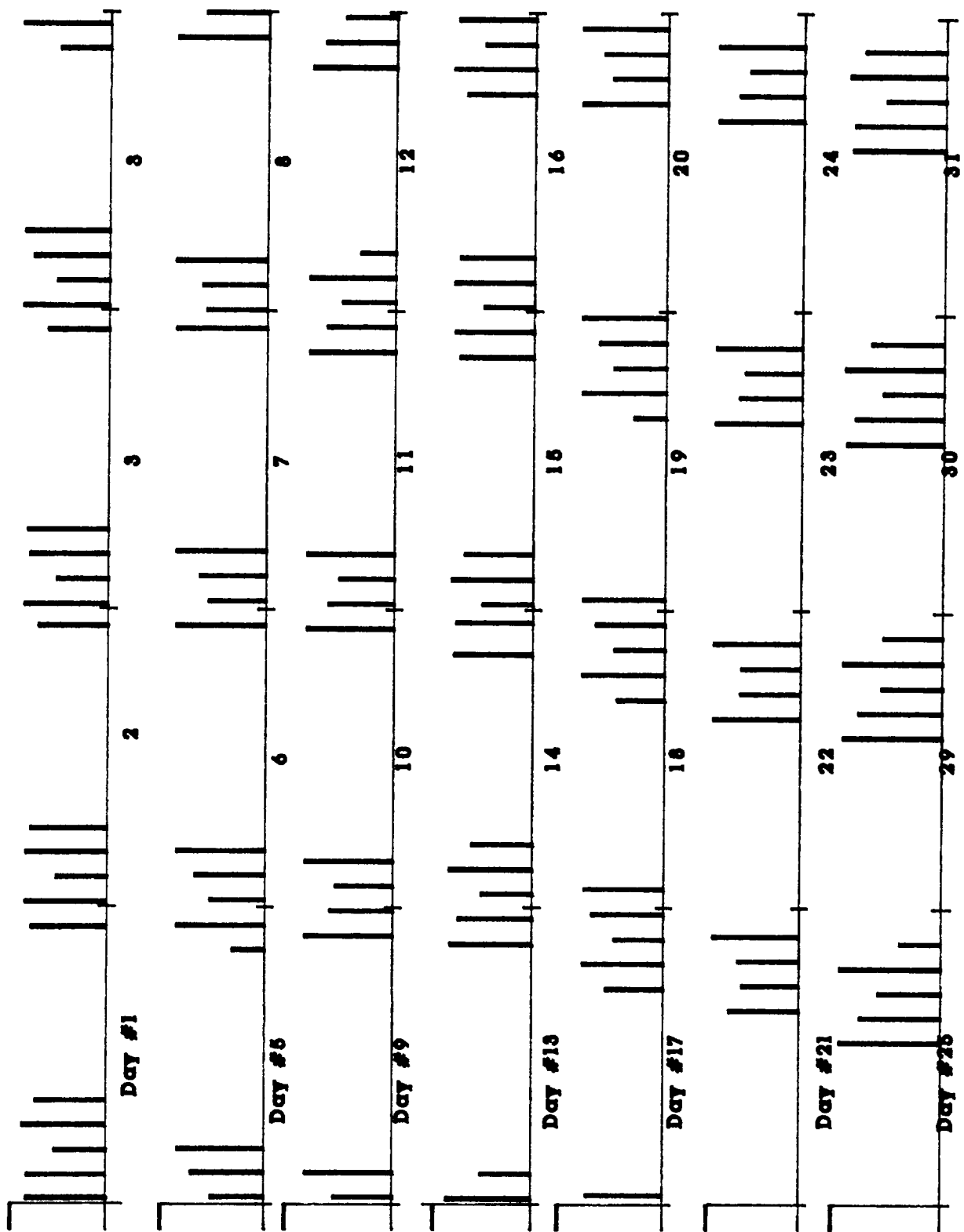


Figure A.7.1 Orbital Pass History

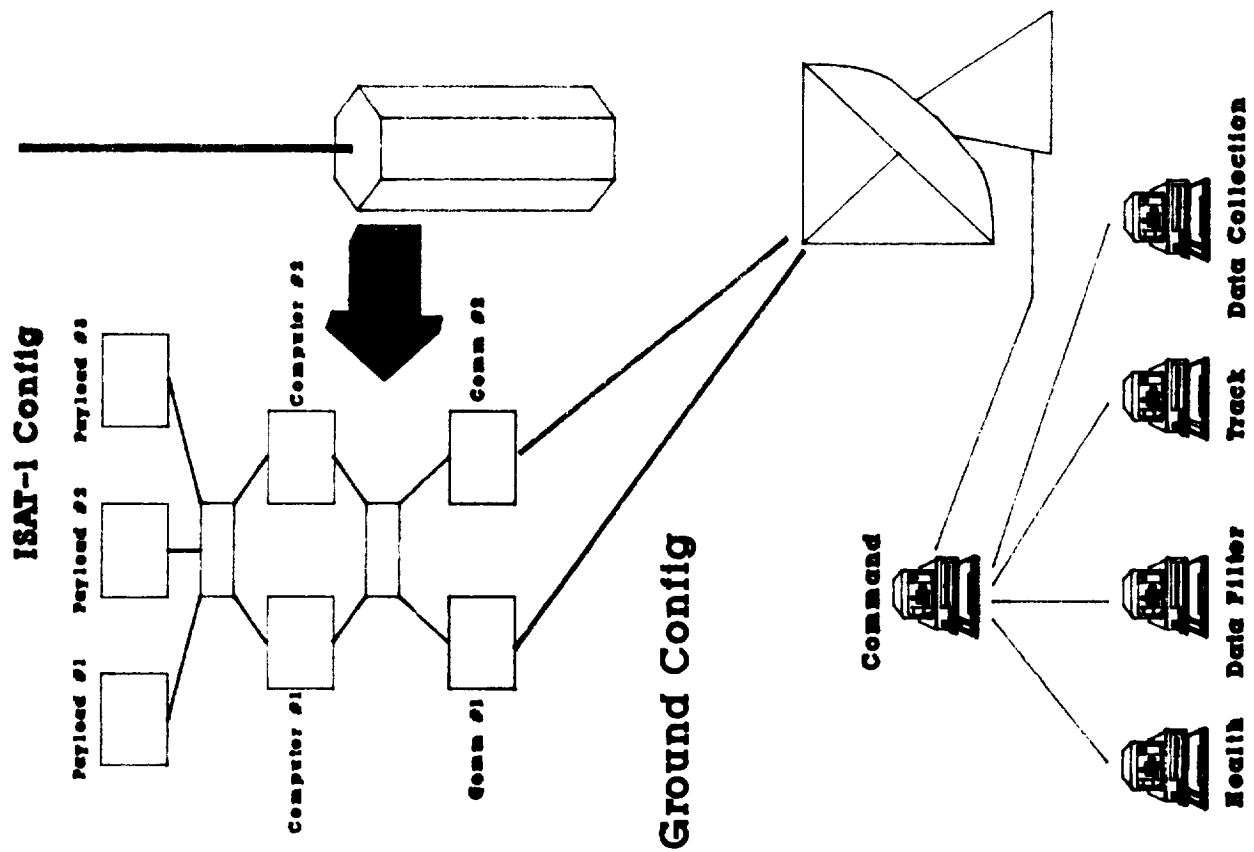


Figure A.7.2 Communications System

Line of Sight Analysis for					
	Altitude = 800 km				
	Inclination = 50 degrees				
Pass	Time of Pass (h:mm)	Duration of Pass (min)	Time Since Last Pass (hrs)	Range Farthest Approach (km)	Range Closest Approach (km)
1	0	11.92	7	3289.9	2319
2	1:42	15.84	1.71	3294.9	969.2
3	3:28	15.97	1.76	3306.5	999.4
4	5:14	15.62	1.78	3316.7	1253.1
5	7:01	16.04	1.77	3321.3	942.4
6	8:46	15.66	1.76	3321	1080.3
7	10:33	10.77	1.78	3314.7	2535.8
8	23:32	11.3	12.97	3290.1	2441.4
9	1:14	15.75	1.71	3294.2	1029.3
10	3:00	16.01	1.76	3305.7	966.9
11	4:46	15.62	1.78	3316.2	1254.4
12	6:33	16	1.77	3321.2	974.7
13	8:18	15.77	1.76	3321.2	1014.1
14	10:05	11.46	1.78	3315.6	2411.8
15	23:04	10.59	12.99	3290.4	2565.9
16	0:46	15.63	1.7	3293.6	1097.4
17	2:32	16.04	1.76	3304.8	934.8
18	4:18	15.62	1.78	3315.6	1252.3
19	6:04	15.96	1.77	3321	1007.2
20	7:50	15.87	1.76	3321.4	956.1
21	9:37	12.06	1.78	3316.3	2289.9
22	22:37	9.76	13	3291	2692.2
23	0:18	15.49	1.69	3293	1172.7
24	2:03	16.07	1.75	3303.9	904.1
25	3:50	15.63	1.78	3315	1246.9
26	5:36	15.93	1.77	3320.8	1039.3

27				7:22		15.94		1.76	3321.5		907.1
28				9:08		12.59		1.77	3317		2170.3
29				22:09		8.8		13.01	3291.7		2820.2
30				23:50		15.33		1.69	3292.4		1254.6
31				1:35		16.1		1.75	3303		875.8
32				3:22		15.65		1.78	3314.3		1238.2
33				5:08		15.89		1.77	3320.6		1070.4
34				6:54		16.01		1.76	3321.6		867.8
35				8:40		13.06		1.77	3317.6		2053.1
36				21:42		7.62		13.02	3292.8		2949.8
37				23:22		15.14		1.68	3291.9		1342.3
38				1:07		16.11		1.75	3302.2		850.8
39				2:54		15.67		1.77	3313.6		1226.2
40				4:40		15.85		1.77	3320.3		1100.1
41				6:26		16.06		1.76	3321.6		838.5
42				8:12		13.48		1.77	3318.1		1938.5
43				21:14		6.12		13.04	3294.2		3080.8
44				22:54		14.93		1.66	3291.4		1435.2
45				0:39		16.13		1.74	3301.3		830.6
46				2:25		15.69		1.77	3312.9		1211.1
47				4:12		15.81		1.77	3320		1127.8
48				5:58		16.09		1.76	3321.7		819.4
49				7:44		13.86		1.77	3318.6		1826.9
50				20:48		3.92		13.06	3296.6		3213.2
51				22:26		14.7		1.65	3290.9		1532.8
52				0:11		16.13		1.74	3300.4		816.5
53				1:57		15.72		1.77	3312.2		1193
54				3:44		15.77		1.78	3319.7		1153.4
55				5:30		16.11		1.76	3321.7		810.3
56				7:16		14.19		1.77	3319		1718.5
57				21:58		14.43		14.71	3290.6		1634.7
58				23:43		16.13		1.74	3299.6		809.6
59				1:29		15.75		1.77	3311.5		1172
60				3:16		15.74		1.78	3319.4		1176.5
61				5:01		16.13		1.77	3321.7		810.4
62				6:47		14.49		1.77	3319.4		1613.6

63			21:31	14.13	14.72	3290.3	1740.3
64			23:15	16.11	1.73	3298.8	811.3
65			1:01	15.78	1.77	3310.7	1148.4
66			2:47	15.71	1.78	3319	1196.9
67			4:33	16.13	1.77	3321.7	818.8
68			6:19	14.75	1.76	3319.8	1512.6
69			8:09	4.47	1.82	3306.8	3186.2
70			21:03	13.79	12.9	3290	1849.4
71			22:47	16.09	1.73	3298	822.5
72			0:33	15.82	1.77	3309.9	1122.4
73			2:19	15.68	1.78	3318.6	1214.4
74			4:05	16.12	1.77	3321.7	834.3
75			5:51	14.98	1.76	3320.1	1415.9
76			7:40	6.46	1.81	3309.3	3054
77			20:35	13.4	12.92	3289.9	1961.6
78			22:18	16.05	1.73	3297.2	843.6
79			0:04	15.86	1.77	3309.1	1094.2
80			1:51	15.66	1.78	3318.2	1228.9
81			3:37	16.11	1.77	3321.6	855.6
82			5:23	15.18	1.76	3320.4	1324
83			7:11	7.88	1.8	3311.1	2923.3
84			20:07	12.97	12.94	3289.8	2076.7
85			21:50	16	1.72	3296.4	875
86			23:36	15.9	1.76	3308.3	1064.2
87			1:23	15.64	1.78	3317.7	1240.2
88			3:09	16.09	1.77	3321.5	881.3
89			4:55	15.36	1.76	3320.6	1237.4
90			6:42	9.01	1.79	3312.5	2794
91			19:39	12.49	12.95	3289.8	2194.4
92			21:22	15.93	1.72	3295.6	916.3
93			23:08	15.93	1.76	3307.4	1032.8
94			0:55	15.63	1.78	3317.2	1248.2
95			2:41	16.07	1.77	3321.4	910.2
96			4:27	15.52	1.76	3320.8	1156.8
97			6:14	9.94	1.79	3313.7	2666.3
98			19:12	11.94	12.96	3289.9	2314.5

99			20:54	15.85	1.71	3294.9	967.2
100			22:40	15.97	1.76	3306.6	1000.6
101			0:27	15.62	1.78	3316.7	1253
102			2:13	16.04	1.77	3321.3	941.3
103			3:59	15.65	1.76	3321	1082.9
104			5:45	10.74	1.78	3314.6	2540.4
105			18:44	11.32	12.97	3290.1	2436.9
106			20:26	15.75	1.71	3294.2	1026.9
107			22:12	16.01	1.76	3305.7	968.1
108			23:58	15.62	1.78	3316.2	1254.4
109			1:45	16	1.77	3321.2	973.5
110			3:30	15.77	1.76	3321.2	1016.4
111			5:17	11.43	1.78	3315.5	2416.3
112			18:16	10.62	12.98	3290.4	2561.2
113			19:58	15.63	1.7	3293.6	1094.7
114			21:44	16.04	1.76	3304.8	936
115			23:30	15.62	1.78	3315.6	1252.4
116			1:16	15.97	1.77	3321	1006
117			3:02	15.86	1.76	3321.3	958.1
118			4:49	12.04	1.78	3316.3	2294.4
119			17:49	9.8	13	3291	2687.5
120			19:30	15.49	1.69	3293	1169.8
121			21:15	16.07	1.75	3303.9	905.2
122			23:02	15.63	1.78	3315	1247.1
123			0:48	15.93	1.77	3320.8	1038.1
124			2:34	15.94	1.76	3321.5	908.7
125			4:21	12.57	1.77	3317	2174.6
126			17:21	8.83	13.01	3291.7	2815.4
127			19:02	15.33	1.69	3292.4	1251.5
128			20:47	16.1	1.75	3303.1	876.8
129			22:34	15.65	1.78	3314.3	1238.5
130			0:20	15.89	1.77	3320.6	1069.3
131			2:06	16.01	1.76	3321.6	869
132			3:52	13.05	1.77	3317.6	2057.4
133			16:54	7.67	13.02	3292.7	2945
134			18:34	15.15	1.68	3291.9	1339

135			20:19	16.11		1.75	3302.2		851.7
136			22:06	15.66		1.77	3313.7		1226.7
137			23:52	15.85		1.77	3320.3		1099
138			1:38	16.05		1.76	3321.6		839.4
139			3:24	13.47		1.77	3318.1		1942.7
140			16:27	6.18		13.04	3294.2		3076
141			18:06	14.94		1.66	3291.4		1431.7
142			19:51	16.13		1.74	3301.3		831.3
143			21:37	15.69		1.77	3313		1211.7
144			23:24	15.81		1.77	3320		1126.9
145			1:10	16.09		1.76	3321.7		819.9
146			2:56	13.84		1.77	3318.6		1831
147			16:00	4.03		13.06	3296.5		3208.3
148			17:38	14.71		1.65	3291		1529.1
149			19:23	16.13		1.74	3300.5		816.9
150			21:09	15.72		1.77	3312.3		1193.7
151			22:56	15.78		1.78	3319.7		1152.5
152			0:42	16.11		1.76	3321.7		810.4
153			2:28	14.18		1.77	3319		1722.5
154			17:11	14.44		14.71	3290.6		1630.8
155			18:55	16.13		1.74	3299.6		809.8
156			20:41	15.75		1.77	3311.5		1172.9
157			22:28	15.74		1.78	3319.4		1175.7
158			0:14	16.13		1.77	3321.7		810.2
159			1:59	14.48		1.77	3319.4		1617.4
160			16:43	14.14		14.72	3290.3		1736.3
161			18:27	16.11		1.73	3298.8		811.1
162			20:13	15.78		1.77	3310.7		1149.3
163			21:59	15.71		1.78	3319		1196.2
164			23:45	16.13		1.77	3321.7		818.4
165			1:31	14.74		1.76	3319.8		1516.3
166			3:21	4.38		1.82	3306.7		3191.1
167			16:15	13.8		12.9	3290		1845.3
168			17:59	16.09		1.73	3298		821.9
169			19:45	15.82		1.77	3309.9		1123.4
170			21:31	15.68		1.78	3318.6		1213.8

171				23:17	16.13		1.77	3321.7	833.6
172				1:03	14.97		1.76	3320.1	1419.4
173				2:52	6.4		1.81	3309.2	3058.9
174				15:47	13.42		12.92	3289.9	1957.4
175				17:31	16.05		1.73	3297.2	842.6
176				19:17	15.86		1.77	3309.1	1095.2
177				21:03	15.66		1.78	3318.2	1228.4
178				22:49	16.11		1.77	3321.6	854.7
179				0:35	15.18		1.76	3320.4	1327.3
180				2:23	7.83		1.8	3311	2928.1
181				15:19	12.99		12.94	3289.8	2072.4
182				17:02	16		1.72	3296.4	873.6
183				18:48	15.89		1.76	3308.3	1065.3
184				20:35	15.64		1.78	3317.8	1239.8
185				22:21	16.09		1.77	3321.6	880.3
186				0:07	15.36		1.76	3320.6	1240.5
187				1:54	8.97		1.79	3312.4	2798.8
188				14:51	12.51		12.95	3289.8	2190
189				16:34	15.93		1.72	3295.7	914.6
190				18:20	15.93		1.76	3307.4	1034
191				20:07	15.63		1.78	3317.3	1248
192				21:53	16.07		1.77	3321.4	909.1
193				23:39	15.51		1.76	3320.8	1159.7
194				1:26	9.91		1.79	3313.6	2671
195				14:24	11.97		12.96	3289.9	2310
196				16:06	15.85		1.71	3294.9	965.1
197				17:52	15.97		1.76	3306.6	1001.8
198				19:39	15.62		1.78	3316.8	1252.9
199				21:25	16.04		1.77	3321.3	940.1
200				23:11	15.65		1.76	3321	1085.5
201				0:58	10.71		1.78	3314.6	2545
202				13:56	11.35		12.97	3290.1	2432.3
203				15:38	15.75		1.706532	3294.3	1024.5

Table A.7.1 Orbit History

Appendix B

Source Codes

Appendix B.2

Structures

PROGRAM ISAT1

```

*****
**      This program performs a structural analysis on a
**      skin and stringer structure.
**      Written for the Iowa Satellite Project (ISAT)
**      Written by: Darby G. Cooper
**      based on code originally written for
**      Aero E 361, Spring 1992 semester
**      instructed by Dr. Vogel
**      thanks to M. Thorp for OSEFB routine
**      Written: 01 April 1993
**              03 April 1993
**
**      version 1: Performs reduced model analysis
**                  assumes all panels same thickness
**                  uses 'massest' shelf as loading condition
**                  assumes all stringers same dimensions
**
*****
**      original code contained safety margin evaluation routines
**      which have been removed for this analysis
**
*****
***      this code:
***      input from file
***      calculates section properties
***      output data from file
***      calculates and outputs to file shear flow
***      evaluates safety in spars
***
*****
**      all calculations performed in: SI - meters, kilograms, seconds
*****
*      VARIABLE LISTING
*      DP=DOUBLE PRECISION
*      I=INTEGER
*
*      IN(I) - the unit to read input data from
*      IYY(DP) - the 2nd moment Iyy
*      IZZ(DP) - the 2nd moment Izz
*      IYZ(DP) - the 2nd moment Iyz
*      YBAR(DP) - the y location of the centroid in input coordinates
*      ZBAR(DP) - the z location of the centroid in input coordinates
*      NUMPTS(I) - the number of data points
*
*      NOTE: the parameters listed below contain the array
*            dimensioning parameters used throughout the program
*      More extensive variable listings may be found in the subroutines
*      themselves.
*
*      parameters for the program (used throughout)
*
*      implicit none
*
#include "insert dir/param.f"

```

```

INTEGER I,J,NSTRNG,IN,NUMPTS,ntask,iwork(20)
integer grow,acc1
DOUBLE PRECISION VY,VZ,MY,MZ,MX,Pz
DOUBLE PRECISION PTMASS(20,12),IYY,IZZ,IYZ,YBAR,ZBAR
double precision stress(20,6),coord(20,6),yldlimt
double precision a(20,20),rhs(20),acopy(20,20)
double precision ind,work(20),rcond,darray(20,20)
double precision nu,exy
DOUBLE PRECISION alldat(20,10)

```

```

COMMON/INPUT,NSTRNG,STRNGR,ckin

```

```

COMMON/LOADS/VY,VZ,MY,MZ,MX,Pz

```

```

common/mat1/Exy

```

```

common/data/alldat

```

```

common/output/ptmass

```

```

** note the above common statement is more than just file output!

```

```

* OBTAIN INPUT FROM FILE

```

```

OPEN(UNIT=1,FILE='../input.dir/ver1in.dat',STATUS='UNKNOWN')

```

```

IR=1

```

```

CALL INPUT(IN)

```

```

close(1)

```

```

* CALCULATE SECTION PROPERTIES

```

```

CALL SECPRP,IYY,IZZ,IYZ,YBAR,ZBAR)

```

```

NUMPTS=NSTRNG

```

```

* OUTPUT DATA TO FILE

```

```

** written to file ../output.dir/section.dat

```

```

CALL OUTPUT(NUMPTS,IYY,IZZ,IYZ,YBAR,ZBAR,PTMASS,VY,VZ,MX,MY,MZ)

```

```

* move data from ptmass to stress array for additional processing
* only copy the coordinates of the elements(col. 1 and 2)

```

```

do 900 i=1,numpts
    coord(i,1)=ptmass(i,1)
    coord(i,2)=ptmass(i,2)
    coord(i,3)=alldat(i,3)
900 continue

```

```

** note: the yield limit has been removed from the stress calc

```

```

* calculate the stress in the stringers

```

```

call axistr(ivy,izz,iyz,numpts,coord,stress)

```

```

* format darray for use by dflow

```

```

do 910 i=1,numpts

```

```

        datray(1,2)=ptmass(1,2)
        datray(1,3)=ptmass(1,3)
**      there is a data handling error somewhere
**      datray col 1,2 is coordinates
**      datray col 3 is mass
**      datray col 4 is skin length btw stringers
**      datray col 5 is skin thickness
**      datray col 6 is skin G value
**      datray col 7 is stress
**      nsrdwise values for first version as all skins are same

        datray(1,4)=0.170
        datray(1,5)=skin(1,1)
        datray(1,6)=skin(1,3)
        datray(1,7)=stress(1,4)
910      continue

*      add first point to the end of datray
*      to complete the "circle"
        datray(numpts+1,1)=ptmass(1,1)
        datray(numpts+1,2)=ptmass(1,2)
        datray(numpts+1,3)=ptmass(1,3)
        datray(numpts+1,4)=ptmass(1,6)
        datray(numpts+1,5)=ptmass(1,4)
        datray(numpts+1,6)=ptmass(1,5)
        datray(numpts+1,7)=stress(1,4)

        call dflow(datray,numpts,arhs,ixv,izv,ivz)

*      set variables for matrix solver

        itask=1
        arow=20
        acol=13

*      copy A matrix to preserve original values

        do 1020 i=1,20
            do 1025 j=1,20
                acopy(i,j)=a(i,j)
1025      continue
1020      continue

*      CHECK THE DATA AT THIS POINT - OUTPUT TO FILE ALLDAT1.DAT

        OPEN(UNIT=3,FILE='../output.dir/amatrix.dat',status='unknown')
        write(3,*) 'This is the data before dgefs, after dflow'

        do 995 i=1,numpts
            write(3,990)(a(i,j),j=1,12)
995      continue

990      format(1x,12d16.5)

        call dgefs(a,arow,acol,rhs,itask,ind,work,1work,rcond)

*      invert signs for sign convention

```

```
do 1055 i=1,10  
  rhs(i)=rhs(i)+1.0d0  
1055 continue  
  
call gout(rhs,datrav,numpnts)  
  
stop  
END
```

```
subroutine areaa(b,t,ar)
```

```
* this subroutine calculates the area for a stringer
* that has the geometry of one corner of a hexagon
* and two rectangles set on two axes 120 degrees
* apart
```

```
* written by D. Cooper 03 April 1993
```

```
* variable listing
```

```
* a = length of the right-hand leg
* b = length of the left-hand leg
* t = thickness of the legs
* l1 = leg 1 of the triangle at center
* c = length of inner face of r-hand leg
* d = length of inner face of l-hand leg
* area1 = area of rectangle portion of leg
* area2 = area of triangle portion of leg
* areat = total area
* areal = left leg area
* arear = right leg area
* ar = return area value
```

```
implicit none
```

```
double precision a,b,t,ar
```

```
double precision area1,arear,c,d,areat
```

```
double precision l1,l2,hypot,area1,area2
```

```
double precision hold,temp,term1,term2,term3,p1
```

```
pi = 4.0d0 * atan(1.0d0)
```

```
* check for equal leg lengths
```

```
if (a.eq.b) then
```

```
* calculate the inner face length of right leg
```

```
hold=30.0d0*(pi/180.0d0)
```

```
l1 = t/tan(hold)
```

```
c = a - l1
```

```
* calculate area of rectangle portion of right leg
```

```
area1 = c * t
```

```
* calculate area of triangle portion of right leg
```

```
area2 = 0.5d0 * t * l1
```

```
* right-hand area is the sum of these
```

```
arear = area1 + area2
```

```
* if .eq., then total area is twice this
```

```
areat=2.0d0 * arear
```

```
ar = areat
```

```
else
```

```
print*, 'Capability of program exceeded'
```

```
stop
```

```
endif
```

/f_lver/m/ugrad/361/cooper/ser0464.din

Page 1

return
end

```
subroutine axistrivy,izz,iyz,limit,coord,stress)
```

```
* This subroutine calculates the axial stress in stringers and spar caps  
* of a two-cell wing structure using the un-symmetric beam bending equation
```

```
* VARIABLE LISTING
```

```
* dp - double precision
```

```
* i - integer
```

```
* ivy(dp) - the 2nd moment about the y axis
```

```
* izz(dp) - the 2nd moment about the z axis
```

```
* iyz(dp) - the 2nd mass moment
```

```
* limit(i) - the number of elements to be processed in stringer
```

```
* coord(dp) - array containing the coordinates of the stringers
```

```
** col Content
```

```
** 1 x coordinate
```

```
** 2 y coordinate
```

```
** 3 area
```

```
** 4 unused
```

```
** 5 unused
```

```
** 6 unused
```

```
* stress(dp) - the output array containing the coordinates and stress value
```

```
** col 1 - x coord (coded as y)
```

```
** col 2 - y coord (coded as z)
```

```
** col 3 - blank (yield limit)
```

```
** col 4 - stress value
```

```
* i(i) - loop control variable
```

```
* y(dp) - temporary variable for y coordinate
```

```
* z(dp) - temporary variable for z coordinate
```

```
* numer1(dp) - the numerator of the first coefficient
```

```
* numer2(dp) - the numerator of the second coefficient
```

```
* denom(dp) - the denominator of both coefficients
```

```
* cof1(dp) - the coefficient of the first term
```

```
* cof2(dp) - the coefficient of the second term
```

```
* NOTE requires the common block LOADS from the calling routine
```

```
* the LOADS block must contain the moments about the y and z axis
```

```
*%include 'insert.din/var_axistr.f'
```

```
double precision ivy,izz,iyz,y,z,numer1,numer2,denom,cof1,cof2
```

```
double precision coord(20,12),stress(20,6)
```

```
integer limit,i
```

```
double precision vy,vz,my,mz,mx,Pz
```

```
COMMON/LOADS/ VY,VZ,MY,MZ,MX,Pz
```

```
numer1=mz*ivy-my*iyz
```

```
numer2=my*izz+mz*iyz
```

```
denom=ivy+izz+ivy*2
```

```
cof1=numer1/denom
```

```
cof2=numer2/denom
```

```
do 100 i=1,limit
```

```
  y=coord(i,1)
```

```
stress(1,1)=x  
stress(1,2)=y  
stress(1,3)=coord(1,3)  
stress(1,4)= -1.0d0*cof1*x+cof2*z+ (pz*coord(1,3))  
100 continue  
  
return  
end
```

```
SUBROUTINE CENTRD(PTMASS,NUMPTS,YBAR,ZBAR,iyy,izz,iyz)
```

```
* This subroutine finds the SECTION centroid given the array
* PTMASS that contains the mass and location of the
* stringers(including skin "dumped" into them)
```

```
* modified 22 MAR 92
** modified for program ISAT1 03 APR 93
```

```
* VARIABLE LISTING
```

```
* R - real
* DP - double precision
* I - integer
```

```
* I(I) - LOOP CONTROL VARIABLE
```

```
* PTMASS(DP) - data array containing x,z locations and mass
```

```
* Y(DP) - the y location of the current point of interest
```

```
* Z(DP) - the z location of the current point of interest
```

```
* MASS(DP) - the mass of the current point of interest
```

```
* MASSUM(DP) - the summation of the mass of the section
```

```
* YSUM(DP) - the sum of the y coordinates of the section * pt mass
```

```
* ZSUM(DP) - the sum of the z coordinates of the section * pt mass
```

```
* YBAR(DP) - the y location of the centroid of the section
```

```
* ZBAR(DP) - the z location of the centroid of the section
```

```
* NUMPTS(I) - the number of data points to process
```

```
* NSTRGS(I) - the number of stringers
```

```
* NCAPS(I) - the number of spar caps
```

```
* VARIABLE INITIALIZATION
```

```
*%include ../insert.dir/var_cd.f
```

```
implicit none
```

```
DOUBLE PRECISION PTMASS(20,3),VY,VZ,MY,MX,MZ
```

```
DOUBLE PRECISION Y,Z,MASS,MASSUM,YBAR,ZBAR,YSUM,ZSUM
```

```
double precision iyy,izz,iyz
```

```
INTEGER I,NUMPTS
```

```
COMMON/LOADS/VY,VZ,MY,MZ,MX
```

```
* CALCULATE THE SUM OF THE PRODUCT OF Y*MASS, Z*MASS, AND SUM OF THE MASSES
```

```
* INITIALIZE SUMMATION VARIABLES TO ZERO
```

```
YSUM=0.000
```

```
ZSUM=0.000
```

```
MASSUM=0.000
```

```
iyy=0.000
```

```
izz=0.000
```

```
iyz=0.000
```

```
* SUM THE NECESSARY TERMS
```

```
DO 600 I=1,NUMPTS
```

```
Y=PTMASS(I,1)
```

```
Z=PTMASS(I,2)
```

```
MASS=PTMASS(I,3)
```

```
MASSUM=MASSUM + MASS
```

```
      ZSUM=ZSUM+Z*MASS
600  CONTINUE

*      CALCULATE CENTROID LOCATION

      YBAR=YSUM/MASSUM
      ZBAR=ZSUM/MASSUM

*      TRANSFORM COORDINATES

      DO 610 I=1,NUMPTS
          PTMASS(I,1)=PTMASS(I,1)-YBAR
          PTMASS(I,2)=PTMASS(I,2)-ZBAR
610  CONTINUE

      do 620 i=1,numpts
          Y=PTMASS(i,1)
          Z=PTMASS(i,2)
          MASS=PTMASS(i,3)
          ivy=ivy+(Z**2)*mass
          izz=izz+(Y**2)*mass
          ivz=ivz+Y*Z*mass
620  continue

**      note: this error was for original problem solution
**      the actual computations will need to be verified
*      multiply ivy,izz by factor of 10
*      multiply ivz by factor of 5
*      A mass factor that should divide out doesn't, i.e. the numerical
*      answers are off by a factor of 10 for ivy and izz and a factor
*      of 5 for ivz.
*      ivy=ivy*10.0d0
*      izz=izz*10.0d0
*      ivz=ivz*5.0d0

      return
      END
```

SUBROUTINE INPUT(IN)

This subroutine reads data from a file opened on IN

The data read is:

- the number of stringers
- properties of the stringers, one line for each stringer
(properties = y coord, z coord, area, E, W)
- properties for the spar caps
(properties = a,b,c,d, E,W, y coord, z coord)
- properties for the spar cap shear web, one line for each web
(properties = t,G,W)
- properties for the skin of each cell, one line for each cell
(properties = t,G,W)
- internal resultant loads
(loads = VY,VZ,MY,MZ,MX)

distinction between stringer and sparcap is not useful for this problem - any variation in stringer properties can be accommodated by the variation of the input data -

PARAMETER LISTING

These parameters may be varied to accomodate various problems

I - integer

R - real

DP - double precision

- MAXSTR(I) - the maximum number of stringers used for array dimensioning
- STRDAT(I) - the number of data items in each row of the stringer data mat
- WEBDAT(I) - the number of data items in each row of the web data matrix
- SKNDAT(I) - the number of data items in each row of the skin cell data matrix
- npans(I) - the number of skin panels

VARIABLE LISTING

I - integer

R - real

DP - double precision

WSTRNG(I) - the number of stringers(read from file)

STRNGR(DP) - 2D array containing stringer data

col. 1 - x coordinate

col. 2 - y coordinate

col. 3 - x dimension

col. 4 - y dimension

col. 5 - thickness

col. 6 - "weight" (actually density per unit area)

SPIN(DP) - 2D array containing skin data

col. 1 - thickness

col. 2 - "weight" (density per unit area)

col. 3 - matl property E

VY(DP) - internal resultant shear load in y-direction

VX(DP) - internal resultant shear load in x-direction

MY(DP) - internal resultant moment about y-axis(bending)

MZ(DP) - internal resultant moment about z-axis(torsion)

MX(DP) - internal resultant moment about x-axis(bending)

IN(I) - the input file unit - passed as an argument

E-W(DP) - young's modulus in the x-y plane

STRNGR is given dimensions (MAXSTR,STRDAT)

```

*
*      include insert file with variable declarations

**      global parameter definition and array dimensioning
      implicit none
%include"../insert.dir/param.f"

**      variable declarations

      INTEGER I,J,NSTRNG,IN
      DOUBLE PRECISION VY,VZ,MY,MZ,MX,Pz
      double precision Exy

      COMMON/INPUT/NSTRNG,STRNGR,skin
      COMMON/LOADS/VY,VZ,MY,MZ,MX,Pz
      common/mat1/Exy

*      NOTE:  the data file MUST be opened in the calling routine

*      read in stringer data
**      note:  the number of data points per stringer is hardwired -
**      it must be changed in source code!

      READ(IN,*)NSTRNG

      DO 100 I=1,NSTRNG
         READ(IN,*)(STRNGR(I,1),STRNGR(I,2),STRNGR(I,3),STRNGR(I,4),
1          STRNGR(I,5),strngr(I,6)
100    CONTINUE

*      read in skin data

      DO 130 J=1,npuntis
         READ(IN,*) skin(J,1),skin(J,2),skin(J,3)
130    CONTINUE

*      read internal loads

      READ(IN,*)(VY,VZ,MY,MZ,MX)

*      read axial shelf load per stringer

      read(in,*)Pz

*      read material property

      read(in,*)Exy

      return
End

```

```

SUBROUTINE OUTPUT(NUMPTS,IYY,IZZ,IYZ,YBAR,ZBAR,PTMASS,VY,VZ,MX
1  MY,MZ)

```

* This subroutine outputs data from the program MAIN1d

```

49include '../insert.dir/var_opt.f'
   implicit none
   DOUBLE PRECISION PTMASS(20,12)
   DOUBLE PRECISION VY,VZ,MX,MY,MZ,YBAR,ZBAR
   DOUBLE PRECISION IYY,IZZ,IYZ
   INTEGER I,J,NUMPTS

   OPEN(UNIT=3,FILE='../output.dir/section.dat',STATUS='UNKNOWN')

   WRITE(3,900)'The centroid location of this wing section is:'
900  FORMAT(1X,A50)

   WRITE(3,910)'Y centroid coordinate: ',YBAR
910  FORMAT(1X,A30,D10.4)
   WRITE(3,910)'Z centroid coordinate: ',ZBAR

   WRITE(3,900)'Internal resultant load summary at the centroid'
   WRITE(3,920)'Shear force in the Y-direction: ',VY
   WRITE(3,920)'Shear force in the Z-direction: ',VZ
   WRITE(3,920)'Moment about the X-axis: ',MX
   WRITE(3,920)'Moment about the Y-axis: ',MY
   WRITE(3,920)'Moment about the Z-axis: ',MZ
   WRITE(3,920)'2nd mass moment, Iyy: ',IYY
   WRITE(3,920)'2nd mass moment, Izz: ',IZZ
   WRITE(3,920)'2nd mass moment, Iyz: ',IYZ

920  FORMAT(1X,A35,D12.6)

*   OUTPUT THE AREAS
   WRITE(3,935)
935  FORMAT(1X)
   WRITE(3,940)'Y','Z','MASS'
940  FORMAT(1D,3A15)

   DO 990 I=1,NUMPTS
       WRITE(3,930)PTMASS(I,1),PTMASS(I,2),PTMASS(I,3)
990  CONTINUE
930  FORMAT(1X,3D15.6)

   CLOSE(3)

END

```

SUBROUTINE POINT(ALLDAT,NUMPTS)

* This subroutine calculates the length of a segment of wing
 * skin. It then calculates the mass of the segment and
 * divides that mass equally between the stringers on either
 * side of the skin segment.
 * It also accounts for the shear webs between the spar caps
 * This subroutine calculates for ONE CELL plus the semi-circular nose
 * section.
 * The array ALLDAT should contain the data for the desired cell, sorted
 * sorted so the stringers AND spar caps are in a clockwise circular fashion
 * around the cell

VARIABLE LISTING

* R - real
 * DP - double precision
 * I - integer

* TEMP1(DP) - temporary value
 * TEMP2(DP) - temporary value
 * Y1(DP) - y coordinate of stringer point 1
 * Y2(DP) - y coordinate of stringer point 2
 * Z1(DP) - z coordinate of stringer point 1
 * Z2(DP) - z coordinate of stringer point 2
 * LENGTH(DP) - length of the skin segment
 * SKNMAS(DP) - skin segment mass
 * ADDMAS(DP) - the amount of mass to be added to each stringer
 * THICK(DP) - the thickness of the skin
 * WEIGHT(DP) - the weight per unit length of the skin
 * G1(DP) - the height of the top flange of spar cap
 * G2(DP) - the height of the top flange of spar cap
 * D1(DP) - the height of the lower flange of the spar cap
 * D2(DP) - the height of the lower flange of the spar cap
 * I(I) - loop control variable
 * J(I) - loop control variable
 * NCELLS(I) - the number of cells
 * NUMPTS(I) - the number of stringers in this cell
 * ALLDAT(DP) - the unified data array

ALLDAT array structure	
COLUMN	Contents
1	y location
2	z location
3	area of stringer(spar)
4	weight/in ² of stringer
5	mass of stringer(spar)
6	"d" dimension of spar(0 for stringer)
7	"d" dimension of spar(0 for stringer)
8	skin thickness
9	weight/in of skin
10	G of skin
11	mass of skin added to stringer
12	new value of point mass equivalent
13	length of skin between stringer i and i+1

VARIABLE DECLARATION

**%include ../insert.dir/var_pm.f

```

INTEGER MAXSTR,UNIDAT
PARAMETER(MAXSTR=20,UNIDAT=15)
DOUBLE PRECISION TEMP1,TEMP2,Y1,Y2,Z1,Z2,SKNMAS,ADDMAS
DOUBLE PRECISION C1,C2,D1,D2,ALLDAT(MAXSTR,UNIDAT)
DOUBLE PRECISION LENGTH,WEIGHT,THICK,PI,C,R
INTEGER I,J,NSTRNG,NUMPTS

```

```

*      handle the square cell
*      note:  the thickness used for a segment is controlled by the skin data
*      associated with the first point
*
DO 400 J=1,NUMPTS

*      ASSIGN GEOMETRIC AND PHYSICAL PROPERTIES TO CONSTANTS FOR MANIPULATI

      Y1=ALLDAT(J,1)
      Y2=ALLDAT(J+1,1)
      Z1=ALLDAT(J,2)
      Z2=ALLDAT(J+1,2)
      C1=ALLDAT(J,6)
      C2=ALLDAT(J+1,6)
      D1=ALLDAT(J,7)
      D2=ALLDAT(J+1,7)
      THICK=ALLDAT(J,8)
      WEIGHT=ALLDAT(J,9)

*      CALCULATE LENGTH, MASS OF SKIN

      TEMP1=(Y1-Y2)**2
      TEMP2=(Z1-Z2)**2
      LENGTH=SQRT(TEMP1+TEMP2)
      alldat(j,13) = length

      SKNMAS=THICK*WEIGHT*LENGTH

*      TAKE HALF OF THE MASS

      ADDMAS = 0.500*SKNMAS

*      ASSIGN THE NEW VALUES TO THE UNIFIED DATA ARRAY
*      NOTE:  ONLY THE JTH ELEMENT IS READY FOR FINAL CALCULATION

      ALLDAT(J,11)=ALLDAT(J,11)+ADDMAS
      ALLDAT(J+1,11)=ALLDAT(J+1,11)+ADDMAS
      ALLDAT(J,12)=ALLDAT(J,11)+ALLDAT(J,5)

400  CONTINUE

*      Note:  The numpts+1 element is the same as the first(j=1) element.
*      The first element should have one-half of the skin mass of the last web
*      added to it also. this is not performed in the loop as only the jth element
*      is "processed" to completion(i.e. one-half of the mass of the skin on BOTH
*      sides of stringer or spar. Therefore, it is necessary to manually add the
*      mass of the skin in this case(the front shear web) to the front upper spar
*      which is the first(j=1) and the last(j=numpts+1) element. After leaving

```

```
alloat(1,12)=alldat(1,12)+addmas
```

```
* CHECK THE DATA AT THIS POINT - OUTPUT TO FILE ALLOAT2.DAT
OPEN(UNIT=3,FILE='../output.dir/alldat2.dat',status='unknown')
write(3,*)'This is the data at the end of point'
write(3,*)'it should have the dumped masses'

do 995 i=1,numpts
    write(3,990)(alldat(i,j),j=1,12)
995 continue

990 format(1x,12d16.5)

close(3)

return
END
```

```
subroutine oflow(datray,numpts,a,rhs,iyy,izz,ivz)
```

```
*
* This subroutine generates the "A" matrix for the equation
* [A]{c} = {RHS}
* which can be solved to find the shear flows in a wing structure.
* This is done by use of numpts-1 nodal equations, one moment equation,
* and one equation from equal angles of twist.
```

VARIABLE LISTING

```
*
*
* db - double precision
* i - integer
```

INPUTS:

```
*
* datray(db) - array with the following structure
* 1 - y coordinate of stringer/spar
* 2 - z coordinate of stringer/spar
* 3 - area of stringer/spar
* 4 - skin length between stringers
* 5 - skin thickness
* 6 - skin G value
* 7 - stress
*
* numpts(i) - the number of elements in datray
* Mx(db) - the moment about the x-axis
```

OUTPUT:

```
*
* A(db) - array with the "A" matrix
* RHS(db) - row vector, the right-hand side of the equation
*
* i(i) - loop control variable
* area(db) - temporary value of area
* abar1(db) - the included area of the nose cell
* abar2(db) - the included area of the rear cell
* sigma(db) - temporary value of stress
* length(db) - temporary value of length
* pi(db) - pi = 3.14159...
* r(db) - the radius of the nose section
* thick(db) - temporary variable for skin thickness
* g(db) - temporary variable for skin G
* b(db) - temporary value for the base of triangles
* h(db) - temporary value for the height of triangles
* abar(db) - value of enclosed section
* temp(db) - temporary value
```

```
*%include'../insert.cin/var_qflow.f'
```

```
IMPLICIT NONE
```

```
double precision datray(20,7),a(20,20),rhs(20),mx,abar,temp
double precision area,abar1,abar2,sigma,length,pi,r,thick,g,b,h
integer i, numpts,j
double precision area1,area2,vy,vz,my,mz,iyy,iyz,izz
double precision k1,k2,denom,numer1,numer2,y,z
double precision len1,len2,len3,s,y2,y3,z2,z3
```

```
COMMON/LOADS/VY,VZ,MY,MZ,MX
```

```
pi = 4.0d0 * atan(1.0)
```

```

*      note:  data for skin is stored at the original con stringer index
*             i.e. the data for the skin between stringer 1 and 2 is stored with
*             stringer 1

*      zero the A matrix
do 890 i=1,20
    do 895 j=1,20
        a(i,j)=0.0d0
895    continue
890  continue

*      calculate constant terms of rhs coefficients

denom=(iyy*izz-(iyz**2))
numer1=(-1.0d0*vy)*(iyy+iyz*vz)
numer2=vz*izz+(-1.0d0*vy)*iyz
k1=-1.0d0*(numer1/denom)
k2=numer2/denom

do 910 i=1,(numpts-1)

*      assign data for handling
y = datray(i,1)
z = datray(i,2)
area = datray(i,3)
sigma = datray(i,7)

*      assign value on main diagonal
a(i,i)=1.0d0

*      check for front spars - they are special case
**      note:  I don't remeber if this condition is
**      problem specific or not - find reference ANSWER!!!!

    if (i.eq.1) then
        a(i,numpts)=-1.0d0
*      a(i,numpts+1)=-1.0d0
    end if

*      assign value
    if (i.ne.1) a(i,i-1)=-1.0d0

*      calculate rhs value

    rhs(i)=area*((k1*y)-(k2*z))*(-10.0d0)+sigma

910  continue

*      establish moment equation

abar2=0.0d0
*      note:  use Hero's thm. to find areas
*      This thm. says area = sqrt(s(s-a)(s-b)(s-c)) where a,b,c are lengths
*      of the sides and s = 0.5(a+b+c)

*      len1 = length of skin between stringers
*      len2 = length from centroid to point i
*      len3 = length from centroid to point i+1

```

```
len1=datray(i,4)
y2=datray(i,1)
z2=datray(i,2)
y3=datray(i+1,1)
z3=datray(i+1,2)
len2 = sqrt(y2**2+z2**2)
len3 = sqrt(y3**2+z3**2)
s = 0.5d0*(len1+len2+len3)
**      note: abs added to prevent small difference neg. values
temp = abs(s*(s-len1)*(s-len2)*(s-len3))
area = sqrt(temp)
abar2=abar2+area
a(numpts,i)=2.0d0*area
*920  continue

*      set moment rhs
*      moment is negative due to shear flow sign convention

rhs(numpts)=mx

*      Establish twist condition

*      for cell 2

*      do 930 i=1,numpts
*          length=datray(i,4)
*          thick=datray(i,5)
*          g=datray(i,6)
*          a(numpts+1,i)=length/(2.0d0*thick*g*abar2)
*930  continue

*      set twist rhs
*      twist rhs is zero - i.e. twist2-twist1 = 0

*      rhs(numpts+1)=0.0d0

return
end
```

```

      subroutine dout(c,datrays,numpts)
*
*      this subroutine outputs the shear flow data to file

*%include'../insert.dir/var_dout.f'

      IMPLICIT NONE

      double precision q(20),datray(20,20)
      integer numpts,i,j,k

*      output results

      open(unit=3,file='../output.dir/qflow.dat',status='unknown')

      write(3,203)'ISAT Shear Flow analysis'
      write(3,203)'origin at geometric center of hexagon'
      write(3,203)'y axis toward launch adapter'
      write(3,203)'x axis to the right, view fr. top'
      write(3,203)'z axis along lenght, postive up'
      write(3,203)'Element 1 is the stringer along x.'
      write(3,203)'Postive shear flow is clock-wise.'
      write(3,203)'

203      format(1x,860)

      write(3,201)'Component','Shear flow','Skin Thickness',
1      'Shear Stress'
      write(3,201)'','(1b/in)','(in)','(psi)'

201      format(1x,4a16)

      do 210 i=1,numpts
         write(3,202)'Element',i,q(i),datray(i,5),datray(i,7)
202      format(1x,a13,i3,3d16,4)
210      continue

*      print out nose section
      write(3,204)'Element',13,q(13),datray(14,5),'N/A'
204      format(1x,a13,i3,2d16,4,a16)

      close(3)

      end

```

SUBROUTINE GEOPR(P, IYY, IZZ, IYZ, YBAR, ZBAR)

* This subroutine calculated the section properties of a wing
* given the input data
* Modified for Problem 10 - 22 MAR 92

* VARIABLE LISTING

* R - real
* DP - double precision
* I - integer
* NCOL(I) - number of columns in a passed array
* NROW(I) - number of rows in a passed array
* OUT1(DP) - output array of a subroutine
* OPTDAT(DP) - the sorted data array
* NUMPTS(I) - the number of points in a data array

* global parameters
#include '../insert/dit/param.f'

INTEGER NROW, NCOL, NUMPTS, I, CTR
DOUBLE PRECISION ALLDAT(MAXSTR, 15), PTMASS(MAXSTR, 10)
DOUBLE PRECISION Y, Z, A, B, C, D, YBAR, ZBAR, yldim1
DOUBLE PRECISION VY, VZ, MY, MZ, MX, MASS, IYY, IZZ, IYZ

COMMON/INPUT/NSTRNG, STRNGR, skin
COMMON/LOADS/VY, VZ, MY, MZ, MX
COMMON/OUTPUT/PTMASS
common/data:alldat

* COMBINE DATA AND SORT INTO CLOCKWISE ORDER

CALL UNIFER(ALLDAT, NSTRNG, STRNGR, skin)

NUMPTS=NSTRNG

* "DUMP" SKIN INTO "STRINGERS" TO CREATE POINT MASSES

CALL POINT(ALLDAT, NUMPTS)

* "TRIM" MATRIX FOR CALCULATION OF SECTION CENTROID

DO 720 I=1, NUMPTS
PTMASS(I, 1)=ALLDAT(I, 1)
PTMASS(I, 2)=ALLDAT(I, 2)
PTMASS(I, 3)=ALLDAT(I, 12)
720 CONTINUE

* CALCULATE CENTROID OF SECTION, TRANSFORM POINT MASS COORDINATES
* CALCULATE SECTION 2ND MASS MOMENTS, IYY, IZZ, IYZ

CALL CENTRD(PTMASS, NUMPTS, YBAR, ZBAR, iyy, izz, iyz)

* TRANSFER LOADS TO CENTROIDAL SYSTEM

MX = MX + VY*YBAR + VZ*ZBAR

* ALL OTHER LOADS UNCHANGED BY COORDINATE TRANSFORM
END

```
SUBROUTINE UNIFER(ALLDAT,NSTRNG,STRNGR,skin)
```

```
* This subroutine creates a unified data matrix of the  
* stringers and skins
```

```
* VARIABLE LISTING
```

```
* R - real  
* DP - double precision  
* I - integer
```

```
* STRNGR(DP) - data array for the stringer
```

```
* skin(DP) - data array for the wing skin
```

```
* ALLDAT(DP)- the unified data array
```

```
* ALLDAT array structure
```

COLUMN	Contents
1	y location
2	z location
3	area of stringer
4	weight/in ² of stringer
5	mass of stringer(spar)
6	"a" dimension of stringer
7	"b" dimension of stringer
8	skin thickness for panel
9	weight/in of skin
10	G of skin
11	mass of skin added to stringer
12	new value of point mass equivalent

```
** note: skin data is controlled by the first index of alldat
```

```
** e.g. alldat(1,8) is the thickness of panel 1, NOT panel 2
```

```
* I(I)- loop control variable
```

```
* NSTRNG(I) - the number of stringers
```

```
* NUMPTS(I) - the number of data points in the current array
```

```
* VARIABLE DECLARATION
```

```
*include'../insert.dir/param.f'
```

```
DOUBLE PRECISION ALLDAT(MAXSTR,15),z
```

```
INTEGER I,J,NSTRNG,NUMPTS,LIMIT,k,nlwr,nupr,ncol
```

```
double precision srt1(20,12),srt2(20,12),upr(20,12),lwr(20,12)
```

```
double precision ar,a,b,t
```

```
* COPY STRINGER x,y,a,b,weight TO ALLDAT
```

```
* CALCULATE area, mass
```

```
DO 500 I=1,NSTRNG
```

```
** x,y coords
```

```
alldat(i,1) = strngr(i,1)
```

```
alldat(i,2) = strngr(i,2)
```

```
** a,b dimension
```

```
a = strngr(i,3)
```

```
b = strngr(i,4)
```

```
t = strngr(i,5)
```

```
** calculate area
```

```
call area(a,b,t,ar)
```

```
** assign to alldat
```

```
alldat(i,6) = a
```

```
alldat(1,3) = ar  
alldat(1,4) = stringr(1,6)  
alldat(1,5) = alldat(1,4) * ar
```

```
500 CONTINUE
```

```
*          ADD SKIN DATA TO ALL STRINGER VALUES
```

```
** note: for version 1 all skin panels are the same -  
** a method needs to be determined for getting the proper  
** skin properties with mid-side stringers when that happens  
** use the first panel entry for now
```

```
do 510 i=1,nstring  
    ALLDAT(I,8)=skin(1,1)  
    ALLDAT(I,9)=skin(1,2)  
    ALLDAT(I,10)=skin(1,3)  
*    SET ADDED SKIN MASS AND POINT MASS VALUE TO ZERO  
    ALLDAT(I,11)=0.000  
    ALLDAT(I,12)=0.000
```

```
510 continue
```

```
* Sort the stringer and spar caps into clock-wise fashion.  
* beginning with the front upper stringer  
** note. this sort worked once - hopefully it still works (DGC 4/3/93)
```

```
* Split upper and lower stringers for sorting  
NUPR=0  
NLWR=0  
NCOL=12
```

```
numpts=nstring+ncaps
```

```
DO 800 I=1,NUMPTS  
    Z=ALLDAT(I,2)  
    IF(Z.GT.0.000)THEN  
        NUPR=NUPR+1  
        DO 810 J=1,NCOL  
            UPR(NUPR,J)=ALLDAT(I,J)  
810        CONTINUE  
    ELSE  
        NLWR=NLWR+1  
        DO 820 J=1,NCOL  
            LWR(NLWR,J)=ALLDAT(I,J)  
820        CONTINUE  
    END IF  
800 CONTINUE
```

```
* SORT UPR INTO ASCENDING ORDER  
  
CALL SORTUP(UPR,NUPR,12,1,SRT1)  
  
* SORT LWR INTO DESCENDING ORDER  
  
CALL SORTDN(LWR,NLWR,12,1,SRT2)
```

```
* REASSEMBLE
```

```
      DO 830 I=1,NUPR
        DO 840 J=1,NCOL
          ALLDAT(I,J)=SRT1(I,J)
840      CONTINUE
830      CONTINUE

      LIMIT=NUPR+NLWR

      DO 850 I=NUPR+1,LIMIT
        J=I-NUPR
        DO 860 K=1,NCOL
          ALLDAT(I,K)=SRT2(J,K)
860      CONTINUE
850      CONTINUE

*      ADD 1ST ELEMENT TO END TO COMPLETE "CIRCLE"

      DO 870 J=1,NCOL
        ALLDAT(LIMIT+1,J)=ALLDAT(1,J)
870      CONTINUE

*      CHECK THE DATA AT THIS POINT - OUTPUT TO FILE ALLDAT1.DAT

      OPEN(UNIT=3,FILE='../output.dir/alldat1.dat',status='unknown')

      write(3,*)'This is the data at the end of UNIFER'
      write(3,*)'it should have all original data sorted'
      write(3,*)'in clockwise order'

      do 995 i=1,numpts
        write(3,990)(alldat(i,j),j=1,12)
995      continue

990      format(1x,12d16.5)

      close(3)

      return
      END

*      NOTE: SPAR CAP COORDINATES ARE THE INPUT COORDINATES NOT
*      THE CENTROIDS OF THE SPAR CAPS
```

ANSYS Finite Element Model

```
/inter,no  
/prep7  
/title, ISAT Structure - Model 2
```

```
et,1,63
```

```
r,1,0.001016,0.001016,0.001016,0.001016  
mp,ex,1,100e6  
mp,dens,1,2.6265e-4
```

```
et,2,63
```

```
r,2,0.002032,0.002032,0.002032,0.002032  
mp,ex,2,100e6  
mp,dens,2,2.6265e-4
```

```
n,1,0.170,0,0  
n,2,0.15725,-0.0220836,0  
n,4,0.09775,-0.1251164,0  
fill  
n,5,0.085,-0.1472  
n,6,0.0595,-0.1472  
n,8,-0.0595,-0.1472  
fill  
n,9,-0.085,-0.1472  
n,10,-0.09775,-0.1257764  
n,12,-0.15725,-0.0220836  
fill  
n,13,-0.170,0  
n,14,-0.15725,0.0220836  
n,16,-0.09775,0.1257764  
fill  
n,17,-0.085,0.1472  
n,18,-0.0595,0.1472  
n,20,0.0595,0.1472  
fill  
n,21,0.085,0.1472  
n,22,0.09775,0.1251164  
n,24,0.15725,0.0220836  
fill
```

```
n,25,0,0,0  
n,26,0.085,0  
n,27,0.0425,-0.0736  
n,28,-0.0425,-0.0736  
n,29,-0.085,0  
n,30,-0.0425,0.0736  
n,31,0.0425,0.0736
```

```
ngen,3,31,1,31,1,0,0,0.0639  
ngen,4,31,63,93,1,0,0,0.0762  
ngen,4,31,156,186,1,0,0,0.063866667  
ngen,2,31,249,279,1,0,0,0.095
```

```
type,2  
real,2
```

```
en,1,1,32,33,2  
egen,9,31,1,1,1
```

en,10,4,35,36,5
en,11,5,36,37,6
egen,9,31,10,11,1

en,30,8,39,40,9
en,31,9,40,41,10
egen,9,31,30,31,1

en,50,12,43,44,13
en,51,13,44,45,14
egen,9,31,50,51,1

en,70,16,47,48,17
en,71,17,48,49,18
egen,9,31,70,71,1

en,90,20,51,52,21
en,91,21,52,53,22
egen,9,31,90,91,1

en,110,24,55,32,1
egen,9,31,110,110,1

type,1
real,1

en,120,2,33,34,3
en,121,3,34,35,4
egen,9,31,120,121,1

en,140,6,37,38,7
en,141,7,38,39,8
egen,9,31,140,141,1

en,160,10,41,42,11
en,161,11,42,43,12
egen,9,31,160,161,1

en,180,14,45,46,15
en,181,15,46,47,16
egen,9,31,180,181,1

en,200,18,49,50,19
en,201,19,50,51,20
egen,9,31,200,201,1

en,220,22,53,54,23
en,221,23,54,55,24
egen,9,31,220,221,1

type,2
real,2

en,240,31,25,26,26
en,241,26,25,27,27
en,242,27,25,28,28
en,243,28,25,29,29
en,244,29,25,30,30
en,245,30,25,31,31
en,246,22,31,26,24
en,247,24,26,1,1

en,248,1,26,2,2
en,249,2,26,27,4,4
en,250,4,27,5,5
en,251,5,27,6,6
en,252,6,27,28,8
en,253,8,28,9,9
en,254,9,28,10,10
en,255,10,28,29,12
en,256,12,29,13,13
en,257,13,29,14,14
en,258,14,29,30,16
en,259,16,30,17,17
en,260,17,30,18,18
en,261,18,30,31,20
en,262,20,31,21,21
en,263,21,31,22,22

egen,2,62,240,263,1
egen,2,93,264,287,1
egen,2,93,288,311,1
egen,2,31,312,335,1

d,80,all,0
d,82,all,0
d,173,all,0
d,175,all,0

f,254,fz,-9
f,275,fz,-9
f,276,fz,-15
f,256,fz,-7.5

f,258,fz,-7.5
f,277,fz,-10.5
f,260,fz,-9

f,264,fz,-1.5
f,278,fz,-10.5
f,265,fz,-3
f,266,fz,-9

f,279,fz,-15
f,268,fz,-7.5
f,270,fz,-7.5
f,274,fz,-10.5
f,249,fz,-3
f,250,fz,-1.5

f,252,fz,-1.5
f,253,fz,-3

f,173,fz,-2.375
f,175,fz,-2.375
f,185,fz,-13.375
f,186,fz,-13.375

f,184,fz,-16
f,183,fz,-11.2
f,182,fz,-11.2
f,181,fz,-16

f,180,fz,-38.2

f,68,fz,-3

f,70,fz,-3

f,72,fz,-1.1

f,74,fz,-1.1

f,76,fz,-1.5

f,78,fz,-2.5

f,79,fz,-2

f,80,fz,-3.75

f,82,fz,-3.75

f,83,fz,-2

f,84,fz,-2.5

f,86,fz,-1.5

f,64,fz,-2.75

f,66,fz,-2.75

f,89,fz,-9.076

f,90,fz,-5.76

f,91,fz,-4.2

f,92,fz,-9.576

f,93,fz,-9.576

f,88,fz,-7.756

f,87,fz,-6

iter,1,1,1

afwrite

finish

/eof

Appendix B.6

Attitude and Control

Dynamic Equations

For a rigid spacecraft the dynamic equations of motion are given by Euler's equation

$$\begin{aligned}\frac{d}{dt}\vec{H} &= -\vec{\omega} \times \vec{H} + \vec{\tau}_d + \vec{\tau}_c \\ \vec{H} &= I\vec{\omega}\end{aligned}$$

where

$$\begin{aligned}\vec{H} &= \text{angular momentum of the spacecraft} \\ I &= \text{moment of inertia tensor of the spacecraft} \\ \vec{\omega} &= \text{spacecraft angular velocity} \\ \vec{\tau}_d &= \text{disturbance torques} \\ \vec{\tau}_c &= \text{control torques}\end{aligned}$$

In the principal axis coordinate system the dynamic equations in component form are

$$\begin{aligned}I_x \frac{d\omega_x}{dt} &= \tau_{dx} + \tau_{cx} + (I_y - I_z)\omega_y\omega_z \\ I_y \frac{d\omega_y}{dt} &= \tau_{dy} + \tau_{cy} + (I_z - I_x)\omega_z\omega_x \\ I_z \frac{d\omega_z}{dt} &= \tau_{dz} + \tau_{cz} + (I_x - I_y)\omega_x\omega_y\end{aligned}$$

Kinematic Equations

The kinematic equations using Euler symmetric parameters can be expressed as follows

$$\frac{d\vec{q}}{dt} = \frac{1}{2}\Omega\vec{q}$$

where

$$\Omega = \begin{bmatrix} 0 & \tilde{\omega}_x & -\tilde{\omega}_y & \tilde{\omega}_z \\ -\tilde{\omega}_x & 0 & \tilde{\omega}_z & \tilde{\omega}_y \\ \tilde{\omega}_y & -\tilde{\omega}_z & 0 & \tilde{\omega}_x \\ -\tilde{\omega}_z & -\tilde{\omega}_y & -\tilde{\omega}_x & 0 \end{bmatrix}$$

and

$$\vec{\omega} = \vec{\omega} + A \begin{bmatrix} 0 \\ -\omega_o \\ 0 \end{bmatrix}$$

where A is the attitude matrix and ω_o is the orbital rate.

The angular rates are related as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ \sin \theta \tan \phi & 1 & -\cos \theta \tan \phi \\ -\sin \theta \sec \phi & 0 & \cos \theta \sec \phi \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

and conversely

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \cos \phi \\ 0 & 1 & \sin \phi \\ \sin \theta & 0 & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Attitude Matrix

The attitude of the spacecraft can be described as a transformation from the nominal reference frame to the body frame. Here, the reference frame is centered at the spacecraft's center-of-mass and is composed of the y-axis parallel to the negative orbit normal, the z-axis toward nadir, and the x-axis normal to both the y-axis and z-axis. For circular orbits, the x-axis is parallel to the spacecraft's velocity vector. The relation between the nominal reference frame, \vec{E} , and the body reference frame, \vec{e} , is

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = A \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

where A is the attitude rotation matrix. Using the yaw-roll-pitch (3-1-2) Euler rotation sequence the attitude matrix, A, can be expressed as

$$A = \begin{bmatrix} \cos \theta \cos \psi - \sin \theta \sin \phi \sin \psi & \cos \theta \sin \psi + \sin \theta \sin \phi \cos \psi & -\sin \theta \cos \phi \\ -\cos \phi \sin \psi & \cos \phi \cos \psi & \sin \phi \\ \sin \theta \cos \psi + \cos \theta \sin \phi \sin \psi & \sin \theta \sin \psi - \cos \theta \sin \phi \cos \psi & \cos \theta \cos \phi \end{bmatrix}$$

where ϕ = roll, θ = pitch, and ψ = yaw.

The Euler angles can be obtained from the elements of the matrix:

$$\begin{aligned} \phi &= \arcsin(a_{23}) & -90^\circ \leq \phi < 90^\circ \\ \theta &= \arctan\left(-\frac{a_{13}}{a_{33}}\right) & 0 \leq \theta < 360^\circ \\ \psi &= \arctan\left(-\frac{a_{21}}{a_{22}}\right) & 0 \leq \psi < 360^\circ \end{aligned}$$

Alternatively, the attitude matrix can be expressed using Euler symmetric parameters q_1, q_2, q_3, q_4 as follows

$$A = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

and conversely, the Euler symmetric parameters can be found from the attitude matrix

$$\begin{aligned} q_4 &= \pm \frac{1}{2}(1 + a_{11} + a_{22} + a_{33})^{\frac{1}{2}} \\ q_1 &= \frac{1}{4q_4}(a_{23} - a_{32}) \\ q_2 &= \frac{1}{4q_4}(a_{31} - a_{13}) \\ q_3 &= \frac{1}{4q_4}(a_{12} - a_{21}) \end{aligned}$$

Gravity-gradient Torque

The gravity-gradient torque τ_{gg} about the center of mass is given by

$$\tau_{gg} = \frac{3\mu}{R_s^3} \left[\hat{R}_s \times (I \cdot \hat{R}_s) \right]$$

where

- μ = Earth's gravitational constant
- \vec{R}_s = geocentric radius vector to the spacecraft's center-of-mass in the body reference frame
- R_s = magnitude of \vec{R}_s
- \hat{R}_s = unit vector of \vec{R}_s
- I = moment-of-inertia tensor

Solar Radiation Torque

The solar radiation torque on the spacecraft, $\vec{\tau}_{solar}$ is the vector sum of all the torques on the individual surfaces generated by a solar radiation force on those surfaces, \vec{F}_i , and is given by

$$\vec{\tau}_{solar} = \sum_{i=1}^n \vec{R}_i \times \vec{F}_i$$

where \vec{R}_i is the vector from the spacecraft center-of-mass to the center-of-pressure of the i th element. The force generated by solar radiation on a flat plane is given by

$$\vec{F} = -PA \cos \theta \left[(1 - C_s) \hat{S} + 2 \left(C_s \cos \theta + \frac{1}{3} C_d \right) \hat{N} \right]$$

where

- P = mean momentum flux acting on a surface normal to the solar radiation
- A = surface area
- \hat{S} = unit vector from the spacecraft to the sun
- \hat{N} = unit outward normal vector of the surface
- θ = angle between \hat{S} and \hat{N}
- C_s = coefficient of specular reflection
- C_d = coefficient of diffuse reflection

Aerodynamic Torque

The aerodynamic torque, $\vec{\tau}_{aero}$, is calculated the same way as the torque due to solar radiation. The aerodynamic torque is calculated by summing the vectors of the torques on individual surfaces generated by the force due to aerodynamic drag, i.e.

$$\vec{\tau}_{aero} = \sum_{i=1}^n \vec{R}_i \times \vec{F}_i$$

where again \vec{R}_i is the vector from the spacecraft center-of-mass to the center-of-pressure of the i th element. The aerodynamic force on a flat plane with area, A , is given by

$$\vec{F} = -\frac{1}{2} C_d \rho V^2 A (\hat{N} \cdot \hat{V}) \hat{V}$$

and the force on a sphere with radius, R , is

$$\vec{F} = -\frac{1}{2} C_d \rho V^2 \pi R^2 \hat{V}$$

where

- C_d = coefficient of drag
- ρ = atmospheric density
- \vec{V} = velocity vector of spacecraft relative to the atmosphere
- \hat{V} = unit vector of \vec{V}
- V = magnitude of \vec{V}
- \hat{N} = unit outward normal vector of plane

Magnetic Torque

The torque due to the spacecraft's magnetic moment is given by

$$\vec{\tau}_m = \vec{m} \times \vec{B}$$

where

- $\vec{\tau}_m$ = magnetic torque acting on the spacecraft ($N \cdot m^2$)
- \vec{m} = spacecraft magnetic moment ($A \cdot m^2$)
- \vec{B} = Earth's magnetic flux density (Wb/m^2)

Earth's Magnetic Field

The Earth's magnetic field, B , can be represented as the gradient of a scalar potential function, V

$$\vec{B} = -\nabla V$$

and V can be represented by a spherical harmonic model

$$V(r, \theta, \phi) = a \sum_{n=1}^k \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^n (g^{n,m} \cos m\phi + h^{n,m} \sin m\phi) P^{n,m}(\theta)$$

where

- r = geocentric distance
- θ = coelevation
- ϕ = east longitude from Greenwich
- a = equatorial radius of the Earth (6371.2 km)
- g and h = coefficients of the IGRF model
- m = order of model
- n = degree of model
- P = Gaussian normalized Legendre function

The field in topocentric coordinates is then

$$\begin{aligned} B_r &= \sum_{n=1}^k \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^n (g^{n,m} \cos m\phi + h^{n,m} \sin m\phi) P^{n,m}(\theta) \\ B_\theta &= -\sum_{n=1}^k \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^n (g^{n,m} \cos m\phi + h^{n,m} \sin m\phi) \frac{\partial P^{n,m}(\theta)}{\partial \theta} \\ B_\phi &= -\frac{1}{\sin \theta} \sum_{n=1}^k \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^n m (-g^{n,m} \sin m\phi + h^{n,m} \cos m\phi) P^{n,m}(\theta) \end{aligned}$$

where

- B_r = radial component, outward positive
- B_θ = coelevation component, South positive
- B_ϕ = azimuthal component, East positive

Various Coordinate System Transformations

The inertial cartesian coordinate system is related to the spherical local tangent coordinate system by

$$\begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} = R \begin{bmatrix} i_r \\ i_\phi \\ i_\theta \end{bmatrix}$$

where R is the orthogonal rotation matrix

$$R = \begin{bmatrix} \sin \phi \cos \theta & \cos \phi \cos \theta & -\sin \theta \\ \sin \phi \sin \theta & \cos \phi \sin \theta & \cos \theta \\ \cos \phi & -\sin \phi & 0 \end{bmatrix}$$

where

$$\begin{aligned} \phi &= \text{coelevation} \\ \theta &= \text{right ascension} \end{aligned}$$

The polar coordinate system is related to the perifocal coordinate system by

$$\begin{bmatrix} i_e \\ i_p \\ i_h \end{bmatrix} = R \begin{bmatrix} i_r \\ i_\theta \\ i_h \end{bmatrix}$$

where R is the orthogonal rotation matrix

$$R = \begin{bmatrix} \cos \nu & -\sin \nu & 0 \\ \sin \nu & \cos \nu & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where ν is the true anomaly

The inertial cartesian coordinate system is related to the perifocal system by

$$\begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} = R \begin{bmatrix} i_e \\ i_p \\ i_h \end{bmatrix}$$

where R is the rotation matrix

$$R = \begin{bmatrix} \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i & -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i & \sin \Omega \sin i \\ \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i & -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i & -\cos \Omega \sin i \\ \sin \omega \sin i & \cos \omega \sin i & \cos i \end{bmatrix}$$

where

$$\begin{aligned} \Omega &= \text{longitude of the ascending node} \\ \omega &= \text{argument of periapsis} \\ i &= \text{inclination} \end{aligned}$$


```

1  PROGRAM Driver
2  .....
3  .....
4  .....
5  Variables:
6  .....
7  .....
8  a = transformation matrix from inertial to S/C body coord.
9  ecc = argument of perapsis of S/C orbit
10 epoch = time of beginning of integration
11 eps = maximum allowable local truncation for integrator
12 gp = gain for position error, vector
13 gd = gain for rate error, vector
14 hain = minimum step size to be taken by integrator
15 htry = initial guess for step size of integrator
16 incln = inclination of S/C orbit
17 ix, iy, iz = principle inertia components
18 lasec = longitude of ascending node of S/C orbit
19 mu = gravitational constant of earth
20 msat = saturation dipole of torque rods, vector
21 norbit = number of orbits for integration
22 period = period of S/C orbit
23 pitch0 = initial pitch angle
24 peax = maximum allowable pitch angle
25 prate0 = initial pitch rate
26 q1 - q4 = quaternions
27 roll0 = initial roll angle
28 rmax = maximum allowable roll angle
29 rrate0 = initial roll rate
30 rtod = radians to degrees conversion factor
31 rvect = orbital position vector of S/C
32 scale = scale factor for torque rod, vector
33 semi = semi-major axis of S/C orbit
34 tranom = true anomaly of S/C orbit at epoch
35 y = integration variables, vector
36 yaw0 = initial yaw angle
37 ymax = maximum allowable yaw angle
38 yrate = initial yaw rate
39 yvect = orbital velocity vector of S/C
40 .....
41 .....
42 .....
43 .....
44 Input files required:
45 .....
46 sc.in : spacecraft physical parameters
47 orbit.in : orbit parameters
48 initial.in : initial attitude configuration, deadband angles,
49 integration parameters
50 igrf75.in : geomagnetic field model Schmidt normalized
51 coefficients
52 .....
53 Output files:
54 .....
55 att.dat : position attitude history
56 orbit.dat : orbit position history
57 control.dat : control history
58 mag.dat : magnetic field in s/c coordinates history
59 .....
60 IMPLICIT NONE
61 .....
62 .....
63 INTEGER n, m, k, kmax
64 .....
65 REAL*8 ix, iy, iz, msat, scale, gp, gd
66 REAL*8 mu, semi, ecc, incln, lasec, aper, tranom, eanow, mean0
67 REAL*8 motion, period, epoch, green0, green, r, w0
68 REAL*8 roll0, pitch0, yaw0
69 REAL*8 rrate0, prate0, yrate0, w10
70 REAL*8 rmax, pmax, ymax, norbit, hain, htry, eps, dseav

```

```

71 REAL*8 g1, h1, g, h
72 REAL*8 a, q1, q2, q3, q4
73 REAL*8 y, t0, tf
74 REAL*8 rtod, twopi
75 .....
76 PARAMETER (kmax = 8, rtod = 57.29578d0, twopi = 6.283185308d0)
77 .....
78 DIMENSION a(3, 3), g(0:kmax, 0:kmax), g1(0:kmax, 0:kmax)
79 DIMENSION h(0:kmax, 0:kmax), h1(0:kmax, 0:kmax), y(7)
80 DIMENSION msat(3), gp(3), gd(3), scale(3), w10(3)
81 COMMON /inertia/ ix, iy, iz
82 COMMON /orbit/ mu, semi, ecc, incln, lasec, aper, tranom,
83 & motion, mean0, period
84 COMMON /geomat/ a
85 COMMON /geom/ g, h, k
86 COMMON /rods/ msat, scale
87 COMMON /time/ epoch, green0, green
88 COMMON /attmax/ rmax, pmax, ymax
89 COMMON /gains/ gp, gd
90 .....
91 EXTERNAL Deriv, Rkqc, Out
92 .....
93 .....
94 .....
95 ** input files
96 .....
97 OPEN (unit = 5, file = 'igrf75.in', status = 'old')
98 OPEN (unit = 6, file = 'sc.in', status = 'old')
99 OPEN (unit = 7, file = 'orbit.in', status = 'old')
100 OPEN (unit = 8, file = 'initial.in', status = 'old')
101 .....
102 ** output files
103 .....
104 OPEN (unit = 9, file = 'att.dat', status = 'unknown')
105 OPEN (unit = 10, file = 'rate.dat', status = 'unknown')
106 OPEN (unit = 11, file = 'control.dat', status = 'unknown')
107 OPEN (unit = 12, file = 'mag.dat', status = 'unknown')
108 OPEN (unit = 13, file = 'ratem.dat', status = 'unknown')
109 .....
110 ** get S/C parameters
111 .....
112 READ(6, *) ix, iy, iz
113 READ(6, *) msat(1), scale(1)
114 READ(6, *) msat(2), scale(2)
115 READ(6, *) msat(3), scale(3)
116 READ(6, *) gp(1), gd(1)
117 READ(6, *) gp(2), gd(2)
118 READ(6, *) gp(3), gd(3)
119 CLOSE (unit = 6)
120 .....
121 ** get orbital elements
122 .....
123 READ(7, *) mu
124 READ(7, *) semi
125 READ(7, *) ecc
126 READ(7, *) incln
127 READ(7, *) lasec
128 READ(7, *) aper
129 READ(7, *) tranom
130 READ(7, *) epoch
131 READ(7, *) green0
132 CLOSE (unit = 7)
133 .....
134 incln = incln / rtod
135 incln = MOD(MOD(incln, twopi) + twopi, twopi)
136 lasec = lasec / rtod
137 lasec = MOD(MOD(lasec, twopi) + twopi, twopi)
138 aper = aper / rtod

```

```

139 aper = MOD(MOD(aper, twopi) + twopi, twopi)
140 tranom = tranom / rtod
141 tranom = MOD(MOD(tranom, twopi) + twopi, twopi)
142 motion = SQRT(mu / semi**3)
143 eanom = SORT((1.d0 - ecc) / (1.d0 + ecc)) * TAN(tranom / 2.d0)
144 eanom = 2.d0 * ATAN(eanom)
145 mean0 = eanom - ecc * SIN(eanom)
146 mean0 = MOD(MOD(mean0, twopi) + twopi, twopi)
147 period = twopi / motion
148 green0 = green0 / rtod
149
150 ** get initial conditions and integration parameters
151
152 READ(8, *) roll0, pitch0, yaw0
153 READ(8, *) rrate0, prate0, yrate0
154 READ(8, *) rmax, pmax, ymax
155 READ(8, *) norbit
156 READ(8, *) hmain, htry
157 READ(8, *) eps, dxsav
158 READ(8, *) k
159 rmax = rmax / rtod
160 pmax = pmax / rtod
161 ymax = ymax / rtod
162 roll0 = roll0 / rtod
163 pitch0 = pitch0 / rtod
164 yaw0 = yaw0 / rtod
165 rrate0 = rrate0 / rtod
166 prate0 = prate0 / rtod
167 yrate0 = yrate0 / rtod
168
169 ** get geomagnetic field model coefficients
170
171 DO n = 1, kmax
172   DO m = 0, n
173     READ(5, *) g1(n, m), h1(n, m)
174     g1(n, m) = g1(n, m) * 1.d-09
175     h1(n, m) = h1(n, m) * 1.d-09
176   END DO
177 END DO
178 CLOSE (unit = 5)
179 CALL Gheconv (g1, h1, g, h)
180
181 ** calculate initial direction cosine matrix
182
183 CALL Dircos (roll0, pitch0, yaw0, a)
184
185 ** calculate initial Euler parameters
186
187 CALL Qdir (a, q1, q2, q3, q4)
188
189 ** calculate initial orbital angular rate
190
191 r = semi * (1.d0 - ecc**2) / (1.d0 + ecc * COS(tranom))
192 w0 = SQRT(mu / r**3)
193
194 ** set initial conditions for integration
195
196 CALL Rate2 (roll0, pitch0, rrate0, prate0, yrate0, w10)
197 y(1) = w10(1) + a(1, 2) * w0
198 y(2) = w10(2) + a(2, 2) * w0
199 y(3) = w10(3) + a(3, 2) * w0
200 y(4) = q1
201 y(5) = q2
202 y(6) = q3
203 y(7) = q4
204
205 ** do the integration
206
207 t0 = 0.d0

```

```

208 tf = norbit * period
209 dxsav = dxsav * period
210
211 *
212 CALL Odeint (y, 7, t0, tf, eps, htry, hmain,
213             & Deriv, Rkqc, Out, 1, dxsav)
214
215 *
216 CLOSE (unit = 8)
217 CLOSE (unit = 9)
218 CLOSE (unit = 10)
219 CLOSE (unit = 11)
220 CLOSE (unit = 12)
221 CLOSE (unit = 13)
222
223 *
224 END

```

```

1  SUBROUTINE Kepler (mean, e, tol, tr, ierr)
2  .....
3  *
4  * Written by Todd Kuper
5  * September 30, 1992
6  *
7  * This routine will solve the Kepler problem for the eccentric
8  * anomaly using Newton's method given the mean anomaly and eccentricity.
9  * The eccentric anomaly will then be converted to the true anomaly.
10 .....
11 .....
12 .....
13 * Variables:
14 * mean = mean anomaly, input
15 * e = eccentricity, input
16 * tol = absolute tolerance, input
17 * tr = true anomaly, output
18 * ierr = error flag, <0 no convergence, >0 converged, output
19 * ecc = eccentric anomaly
20 * i = iteration counter
21 * imax = maximum number of iterations allowed
22 .....
23 .....
24 IMPLICIT NONE
25 INTEGER i, ierr, imax
26 REAL*8 c, e, ecc, mean, mean1, tol, tr, twopi
27 PARAMETER (imax = 50, twopi = 6.28318530717958640)
28 .....
29 IF ((mean.LT. 0.d0) .OR. (mean.GT. twopi)) THEN
30   mean = MOD(MOD(mean, twopi) + twopi, twopi)
31 END IF
32
33 ** Initialize counter
34
35   i = 0
36   ierr = 1
37
38 ** set initial guess
39
40   ecc = mean + (e * SIN(mean)) / (1.d0 - SIN(mean + e) * SIN(mean))
41
42 ** Iterate using Newton's method
43
44   1 mean1 = ecc - e * SIN(ecc)
45     IF (ABS(mean - mean1) .LT. tol) THEN
46       c = SQRT((1.d0 + e) / (1.d0 - e))
47       tr = 2.d0 * ATAN(c * TAN(ecc / 2.d0))
48       IF ((tr.LT. 0.d0) .OR. (tr.GT. twopi)) THEN
49         tr = MOD(MOD(tr, twopi) + twopi, twopi)
50       END IF
51       RETURN
52     END IF
53     i = i + 1
54     IF (i.LT. imax) THEN
55       ecc = ecc + (mean - mean1) / (1.d0 - e * COS(ecc))
56       GO TO 1
57     END IF
58
59 ** failed to converge, so set error flag
60
61   ierr = -1
62
63   END

```

```

1  SUBROUTINE Magnet (g, h, k, phi, theta, r, alphag, b, bmag)
2  .....
3  *
4  * Written by Todd Ruper
5  * on February 1, 1993
6  * Last modified on Feb. 1, 1993
7  *
8  * This routine will calculate the geomagnetic field at a specified point
9  * above Earth's surface using a spherical harmonic model.
10 .....
11 * I/O parameters:
12 * g, h = vectors containing the Gauss normalized Gaussian
13 * coefficients up to degree 8, T, input
14 * k = degree routine should calculate field to, input
15 * phi = East longitude from Greenwich, radians, input
16 * theta = geocentric distance, km, input
17 * r = geocentric distance, km, input
18 * alphag = right ascension of Greenwich, radians, input
19 * b = vector of field in spherical coordinates, T, output
20 * b(1) = radial direction
21 * b(2) = coelevation direction (South)
22 * b(3) = longitude direction (East)
23 * bmag = magnitude of field, T, output
24 .....
25 *
26 * IMPLICIT NONE
27 *
28 * INTEGER k, kmax, n, m
29 *
30 *
31 * REAL*8 g, h, phi, theta, r, b, bmag, a
32 * REAL*8 km, p, ctheta, stheta, dpdt, cmphi, smphi, cphi, sphi
33 * REAL*8 br2, bt2, bp2, coef, br1, btl, bpl, temp, br, bt, bp
34 * REAL*8 delta, cdel, edel, alpha, calpha, salpha, alphag, halfpi
35 *
36 * PARAMETER (a = 6371.2d0, kmax = 8, halfpi = 1.570796327d0)
37 *
38 * DIMENSION b(3), cmphi(0:kmax), dpdt(0:kmax), h(0:kmax)
39 * DIMENSION g(0:kmax), kmax(0:kmax), h(0:kmax), 0:kmax)
40 * DIMENSION km(0:kmax), 0:kmax), p(0:kmax), 0:kmax)
41 * DIMENSION smphi(0:kmax)
42 *
43 * .....
44 * ctheta = COS(theta)
45 * stheta = SIN(theta)
46 *
47 * .. calculate k
48 *
49 * DO n = 1, k
50 *   IF (n.EQ.1) THEN
51 *     km(n, m) = 0.d0
52 *   ELSE
53 *     km(n, m) = DBLE((n - 1.d0)**2 - m**2)
54 *     / DBLE((n + n - 1) * (n + n - 3))
55 *   END IF
56 *   END DO
57 *
58 * .. calculate p recursively
59 *
60 * p(0, 0) = 1.d0
61 * DO n = 1, k
62 *   DO m = 0, n - 1
63 *     p(n, m) = ctheta * p(n - 1, m) - km(n, m) * p(n - 2, m)
64 *   END DO
65 *   p(n, n) = stheta * p(n - 1, n - 1)
66 * END DO
67 *
68 *
69

```

```

70 * .. calculate partial derivative of p with respect to theta recursively
71 *
72 * dpdt(0, 0) = 0.d0
73 * DO n = 1, k
74 *   dpdt(n, m) = ctheta * dpdt(n - 1, m) - stheta * p(n - 1, m)
75 *   * dpdt(n, m) + km(n, m) * dpdt(n - 2, m)
76 *   *
77 *   END DO
78 *   dpdt(n, n) = stheta * dpdt(n - 1, n - 1)
79 *   * ctheta * p(n - 1, n - 1)
80 *   *
81 *   END DO
82 *
83 * .. calculate COS(m * phi) and SIN(m * phi) recursively
84 *
85 * cmphi(0) = 1.d0
86 * smphi(0) = 0.d0
87 * cphi = COS(phi)
88 * sphi = SIN(phi)
89 * DO m = 1, k
90 *   cmphi(m) = cmphi(m - 1) * cphi - smphi(m - 1) * sphi
91 *   smphi(m) = smphi(m - 1) * cphi + cmphi(m - 1) * sphi
92 *   *
93 *   END DO
94 *
95 * .. calculate components of b in spherical coordinates
96 *
97 * br2 = 0.d0
98 * bt2 = 0.d0
99 * bp2 = 0.d0
100 * DO n = 1, k
101 *   coef = (a / r)**(n + 2)
102 *   br1 = 0.d0
103 *   bt1 = 0.d0
104 *   bp1 = 0.d0
105 *   DO m = 0, n
106 *     temp = g(n, m) * cmphi(m) + h(n, m) * smphi(m)
107 *     br1 = br1 + temp * p(n, m)
108 *     temp = DBLE(m) * (-g(n, m) * smphi(m) + h(n, m) * cmphi(m))
109 *     bp1 = bp1 + temp * p(n, m)
110 *   END DO
111 *   br2 = br2 + (coef * DBLE(n + 1) * br1)
112 *   bt2 = bt2 + (coef * bt1)
113 *   bp2 = bp2 + (coef * bp1)
114 *   END DO
115 *   b(1) = br2
116 *   b(2) = -bt2
117 *   b(3) = -bp2 / stheta
118 *   bmag = SQRT(b(1)**2 + b(2)**2 + b(3)**2)
119 *   END
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 * SUBROUTINE Ghoconv (g1, h1, g, h)
128 * .....
129 * * Written by Todd Ruper
130 * * on February 1, 1993
131 * * Last modified on February 1, 1993
132 *
133 * * This routine will convert the Schmidt normalized Gaussian
134 * * coefficients of the IGRF model to Gauss normalized coefficients.
135 * .....
136 * * I/O parameters
137 * * g1, h1 = vectors containing Schmidt normalized coefficients, input
138 * * g, h = vectors containing Gauss normalized coefficients, output

```

3

```

139 .....
140 IMPLICIT NONE
141 INTEGER k, kronec, m, n
142 REAL*8 g, g1, h, h1, s
143 PARAMETER (k = 8)
144 DIMENSION s(0:k, 0:k), g1(0:k, 0:k), h(0:k, 0:k), h1(0:k, 0:k)
145 DIMENSION s(0:k, 0:k)
146 .....
147 *
148 ** calculate s using recursion
149 *
150 s(0, 0) = 1.d0
151 DO n = 1, k
152   s(n, 0) = s(n - 1, 0) * DBLE(n + n - 1) / DBLE(n)
153   DO m = 1, n
154     IF (m .EQ. 1) THEN
155       kronec = 1
156     ELSE
157       kronec = 0
158     END IF
159     s(n, m) = s(n, m - 1) * SQRT(DBLE((n - m + 1) * (kronec + 1)))
160     *
161     END DO
162   END DO
163 *
164 ** convert Schmidt coefficients to Gaussian coefficients
165 *
166 DO n = 1, k
167   DO m = 0, n
168     g(n, m) = g1(n, m) * s(n, m)
169     h(n, m) = h1(n, m) * s(n, m)
170   END DO
171 END DO
172 *
173 END
174
175

```

```

1 SUBROUTINE Out (y, t, h)
2 .....
3
4 IMPLICIT NONE
5 .....
6
7 REAL*8 mu, semi, ecc, inclin, lasec, aper, tranom, motion
8 REAL*8 mean0, period, w0, r
9 REAL*8 epoch, green0, green
10 REAL*8 a, roll, pitch, yaw, wsc, wscmag, irate, prate, yrate
11 REAL*8 y, t, h, rtod
12 REAL*8 b, bmag, bdot, m
13
14 PARAMETER (rtod = 57.29578d0)
15
16 DIMENSION a(3, 3), b(3), bdot(3), m(3), y(7), wsc(3)
17
18 COMMON /cosmat/ a
19 COMMON /time/ epoch, green0, green
20 COMMON /orbit/ mu, semi, ecc, inclin, lasec, aper, tranom,
21 & motion, mean0, period
22 COMMON /control/ m
23 COMMON /magn/ b, bmag, bdot
24 .....
25 CALL Attit (a, roll, pitch, yaw)
26 roll = roll
27 pitch = pitch
28 yaw = yaw
29 r = semi * (1.d0 - ecc**2) / (1.d0 + ecc * COS(tranom))
30 w0 = SQRT(mu / r**3)
31 wsc(1) = (y(1) - a(1, 2) * w0)
32 wsc(2) = (y(2) - a(2, 2) * w0)
33 wsc(3) = (y(3) - a(3, 2) * w0)
34 wscmag = SQRT(wsc(1)**2 + wsc(2)**2 + wsc(3)**2)
35 CALL Ratel (roll, pitch, wsc, irate, prate, yrate)
36
37 WRITE(9, 10) t/period, roll*rtod, pitch*rtod, yaw*rtod
38 WRITE(10, 12) t/period, wsc(1)*rtod, wsc(2)*rtod, wsc(3)*rtod
39 WRITE(11, 12) t/period, m(1), m(2), m(3)
40 WRITE(12, 12) t/period, b(1), b(2), b(3)
41 WRITE(13, 12) t/period, wscmag * rtod
42
43 10 FORMAT (1x, d14.7, 2x, 3(d10.5, 2x))
44 12 FORMAT (1x, f10.5, 2x, 4(d13.6, 2x))
45 END

```

```

1  SUBROUTINE Odeint (ystart, nvar, x1, x2, eps, h1, hmin,
2  Derivs, Stepr, Out, ip, dxsav)
3  .....
4  IMPLICIT REAL*8 (a-h, o-z)
5  PARAMETER (maxstp = 50000, nmax = 20, tiny = 1.d-30)
6  PARAMETER (tiny2 = 1.d-15)
7  DIMENSION ystart(nvar), yscal(nmax), y(nmax), dydx(nmax)
8  DIMENSION b(3), bold(3), bdot(3)
9  COMMON /magb/ b, bmag, bdot
10 COMMON /oldb/ bold, oldt
11 EXTERNAL Derivs, Stepr, Out
12 .....
13 .....
14 .....
15 ** set-up
16 .....
17 oldt = 0.d0
18 hmini = 1.d10
19 hmaxi = 0.d0
20 x = x1
21 h = SIGN(h1, x2 - x1)
22 kount = 0
23 xsav = x - dxsav * 2.d0
24 DO i = 1, nvar
25   y(i) = ystart(i)
26   END DO
27 .....
28 ** take at most maxstp steps
29 .....
30 DO nstp = 1, maxstp
31   CALL Derivs(y, dydx, x, h)
32 .....
33 ** scaling used to monitor accuracy
34 .....
35 DO i = 1, nvar
36   yscal(i) = 1.d0
37   END DO
38 .....
39 ** save values
40 .....
41 IF (ip .GT. 0) THEN
42   IF (ABS(x - xsav) .GE. ABS(dxsav)) THEN
43     CALL Out (y, x, h)
44     xsav = x
45   END IF
46 .....
47 ** if step can overshoot end, cut down stepsize
48 .....
49 IF ((x + h - x2) * (x + h - x1) .GT. 0.d0) h = x2 - x
50 .....
51 ** save this magnetic vector
52 .....
53 oldt = x
54 bold(1) = b(1)
55 bold(2) = b(2)
56 bold(3) = b(3)
57 .....
58 ** take step
59 .....
60 CALL Stepr(y, dydx, nvar, x, h, eps, yscal,
61 & hdid, hnext, Derivs)
62 .....
63 IF ((x - x2) * (x2 - x1) .GE. 0.d0) THEN
64   DO i = 1, nvar
65     ystart(i) = y(i)
66   END DO
67   IF (ip .GT. 0) THEN
68     CALL Out (y, x, h)
69

```

```

70 END IF
71 RETURN
72 END IF
73 .....
74 IF (ABS(hnext) .LT. hmin) THEN
75   WRITE(*,*) 'Stepsize smaller than minimum'
76   STOP
77 END IF
78 h = hnext
79 hmaxi = MAX(hmaxi, hdid)
80 hmini = MIN(hmini, hdid)
81 END DO
82 .....
83 WRITE(*,*) 'Too many steps'
84 STOP
85 END

```

```

1 SUBROUTINE Orbelm(mu, rv, vv, motion, e, inclin, lanode,
2 & aper, truean, eccan, meanan, p, e)
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 .....
22 .....
23 .....
24 .....
25 .....
26 .....
27 .....
28 .....
29 .....
30 .....
31 .....
32 .....
33 .....
34 .....
35 .....
36 .....
37 .....
38 .....
39 .....
40 .....
41 .....
42 .....
43 .....
44 .....
45 .....
46 .....
47 .....
48 .....
49 .....
50 .....
51 .....
52 .....
53 .....
54 .....
55 .....
56 .....
57 .....
58 .....
59 .....
60 .....
61 .....
62 .....
63 .....
64 .....
65 .....
66 .....
67 .....
68 .....
69 .....

```

Written by Todd Kuper
 February 14, 1992
 This routine will calculate the classical orbital parameters given
 the position vector, velocity vector and the gravitational constant
 Variables:
 mu = gravitational constant, input
 rv = position vector, input
 vv = velocity vector, input
 motion = mean motion, output
 e = magnitude of eccentricity, output
 inclin = inclination angle in radians, output
 lanode = longitude of ascending node in radians, output
 aper = argument of periastron in radians, output
 truean = true anomaly in radians, output
 eccan = eccentric anomaly in radians, output
 meanan = mean anomaly in radians, output
 p = parameter, output
 e = semi-major axis, output
 u = argument of latitude

IMPLICIT NONE
 REAL*8 a, aper, e, eccan, ev(3), h, hv(3)
 REAL*8 inclin, lanode, meanan, motion, mu, n, nv(3)
 REAL*8 p, pi, r, rdotv, rv(3)
 REAL*8 tiny, tmp1, tmp2, truean, twopi, u, v, vv(3)
 INTEGER ip
 COMMON /prrpr/ ip

 .. find magnitude of position and velocity vectors
 r = SQRT(rv(1)**2 + rv(2)**2 + rv(3)**2)
 v = SQRT(vv(1)**2 + vv(2)**2 + vv(3)**2)

 .. find angular momentum vector and magnitude
 hv(1) = rv(2)*vv(3) - vv(2)*rv(3)
 hv(2) = rv(3)*vv(1) - vv(3)*rv(1)
 hv(3) = rv(1)*vv(2) - vv(1)*rv(2)
 h = SQRT(hv(1)**2 + hv(2)**2 + hv(3)**2)
 hv(1) = hv(1)/h
 hv(2) = hv(2)/h
 hv(3) = hv(3)/h

 .. find inclination angle
 inclin = ACOS(hv(3))

 .. find node vector and magnitude
 IF (inclin .LT. tiny .OR. (pi - inclin) .LT. tiny) THEN
 nv(1) = 1.040
 nv(2) = 0.040
 nv(3) = 0.040
 ELSE
 nv(1) = -hv(2)
 nv(2) = hv(1)
 nv(3) = 0.40
 END IF

```

70 .....  

71 .....  

72 .....  

73 .....  

74 .....  

75 .....  

76 .....  

77 .....  

78 .....  

79 .....  

80 .....  

81 .....  

82 .....  

83 .....  

84 .....  

85 .....  

86 .....  

87 .....  

88 .....  

89 .....  

90 .....  

91 .....  

92 .....  

93 .....  

94 .....  

95 .....  

96 .....  

97 .....  

98 .....  

99 .....  

100 .....  

101 .....  

102 .....  

103 .....  

104 .....  

105 .....  

106 .....  

107 .....  

108 .....  

109 .....  

110 .....  

111 .....  

112 .....  

113 .....  

114 .....  

115 .....  

116 .....  

117 .....  

118 .....  

119 .....  

120 .....  

121 .....  

122 .....  

123 .....  

124 .....  

125 .....  

126 .....  

127 .....  

128 .....  

129 .....  

130 .....  

131 .....  

132 .....  

133 .....  

134 .....  

135 .....  

136 .....  

137 .....  

138 .....

```

.. find eccentricity vector and magnitude
 tmp1 = (v*v - mu/r)/mu
 rdotv = rv(1)*vv(1) + rv(2)*vv(2) + rv(3)*vv(3)
 tmp2 = rdotv/mu
 ev(1) = tmp1 + rv(1) - tmp2*vv(1)
 ev(2) = tmp1 + rv(2) - tmp2*vv(2)
 ev(3) = tmp1 + rv(3) - tmp2*vv(3)
 e = SQRT(ev(1)**2 + ev(2)**2 + ev(3)**2)
 IF (e .GE. 1.0d0) THEN
 WRITE(*,*) 'non-elliptical orbit'
 STOP
 END IF

 .. find semi-latus rectum
 p = h*h/mu

 .. find semi-major axis
 a = p/(1.0 - e*e)

 .. find mean motion
 motion = SQRT(mu/a**3)

 .. find longitude of ascending node
 lanode = ATAN2(nv(2), nv(1))
 lanode = MOD(lanode + twopi, twopi)

 .. find true, eccentric, and mean anomalies
 IF (e .LT. tiny) THEN
 truean = 0.0d0
 eccan = 0.0d0
 meanan = 0.0d0
 ELSE
 tmp1 = SQRT((ev(2)*rv(3) - ev(3)*rv(2))**2
 + (ev(3)*rv(1) - ev(1)*rv(3))**2
 + (ev(1)*rv(2) - ev(2)*rv(1))**2)
 truean = ATAN2(tmp1, tmp2)
 IF (rdotv .LT. 0.0d0) THEN
 truean = twopi - truean
 END IF
 truean = MOD(truean + twopi, twopi)
 eccan = SQRT((1.0 - e)/(1.0 + e)) * TAN(truean/2.0d0)
 eccan = 2.0d0 * ATAN(eccan)
 meanan = MOD(eccan + twopi, twopi)
 meanan = eccan - e * SIN(eccan)
 END IF

 .. find argument of latitude
 tmp1 = (hv(2)*nv(3) - hv(3)*nv(2)) + rv(1)
 + (hv(3)*nv(1) - hv(1)*nv(3)) + rv(2)
 + (hv(1)*nv(2) - hv(2)*nv(1)) + rv(3)
 tmp2 = rv(1)*nv(1) + rv(2)*nv(2) + rv(3)*nv(3)
 u = ATAN2(tmp1, tmp2)
 u = MOD(u + twopi, twopi)

 .. find argument of periastron
 aper = u - truean
 aper = MOD(aper + twopi, twopi)
 END

```

139 SUBROUTINE Rvect (tanom, a, e, i, lasec, aper, mu, r, v, rmag)
140 .....
141 * Written by Todd Kuper
142 * September 30, 1992
143 .....
144 * This routine will calculate the position and velocity vectors
145 * in the inertial cartesian coordinate system given the orbital
146 * elements.
147 .....
148 * Variables:
149 * I/O:
150 * tanom = true anomaly, input
151 * a = semi-major axis, input
152 * e = eccentricity, input
153 * i = inclination, input
154 * lasec = longitude of the ascending node, input
155 * aper = argument of periastron, input
156 * mu = gravitational constant, input
157 * r = position vector, output
158 * v = velocity vector, output
159 * rmag = radial distance, output
160 .....
161 * Internal:
162 * alat = argument of latitude
163 * p = semi-latus rectum
164 * h = magnitude of angular momentum
165 .....
166 IMPLICIT NONE
167 REAL*8 a, alat, aper, c, calat, caper, ci, clasc, d1, d2
168 REAL*8 e, h, i, lasec, mu
169 REAL*8 p, r, rmag, salat, saper, slasc, si, tanom, v
170 DIMENSION r(3), v(3)
171 .....
172 ** compute intermediate values
173 alat = aper + tanom
174 p = a * (1.0 - e * e)
175 h = SQRT(p * mu)
176 clasc = COS(lasec)
177 slasc = SIN(lasec)
178 calat = COS(alat)
179 salat = SIN(alat)
180 caper = COS(aper)
181 saper = SIN(aper)
182 ci = COS(i)
183 si = SIN(i)
184 d1 = salat * e + saper
185 d2 = calat * e + caper
186 c = mu / h
187 .....
188 ** compute position vector
189 r(1) = rmag * (clasc * calat - slasc * salat * ci)
190 r(2) = rmag * (slasc * calat + clasc * salat * ci)
191 r(3) = rmag * salat * si
192 .....
193 ** compute velocity vector
194 v(1) = -c * (clasc * d1 + slasc * d2 * ci)
195 v(2) = c * (slasc * d1 - clasc * d2 * ci)
196 v(3) = c * d2 * si
197 END
198
199 .....
200 SUBROUTINE Trans1 (lasc, aper, incl, v1, v2)
201 .....
202 * This routine will transform the perifocal vector, v1 to the inertial
203 * cartesian vector, v2 given the longitude of the ascending node,
204 * argument of periastron and the inclination angle.
205 .....
206 * v1(1) = eccentricity direction
207 * v1(2) = parameter direction
208 * v1(3) = orbit normal direction
209 .....
210 * v2(1) = x direction
211 * v2(2) = y direction
212 * v2(3) = z direction
213 .....
214 IMPLICIT NONE
215 INTEGER i, j
216 REAL*8 lasc, aper, incl, r, v1, v2
217 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
218 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
219 DIMENSION v1(3), v2(3)
220 .....
221 clasc = COS(lasc)
222 slasc = SIN(lasc)
223 caper = COS(aper)
224 saper = SIN(aper)
225 cincl = COS(incl)
226 sinc1 = SIN(incl)
227 .....
228 l1 = clasc * caper - slasc * saper * cincl
229 l2 = -clasc * saper - slasc * caper * cincl
230 l3 = slasc * cincl
231 m1 = slasc * caper + clasc * saper * cincl
232 m2 = -slasc * saper + clasc * caper * cincl
233 m3 = -clasc * sinc1
234 n1 = saper * sinc1
235 n2 = caper * sinc1
236 n3 = cincl
237 .....
238 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
239 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
240 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
241 END
242
243 .....
244 SUBROUTINE Trans2 (lasc, aper, incl, v1, v2)
245 .....
246 * This routine will transform the inertial cartesian vector, v1 to
247 * the perifocal vector, v2 given the longitude of the ascending node,
248 * argument of periastron and the inclination angle.
249 .....
250 * v2(1) = eccentricity direction
251 * v2(2) = parameter direction
252 * v2(3) = orbit normal direction
253 .....
254 * v1(1) = x direction
255 * v1(2) = y direction
256 * v1(3) = z direction
257 .....
258 IMPLICIT NONE
259 INTEGER i, j
260 REAL*8 lasc, aper, incl, r, v1, v2
261 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
262 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
263 DIMENSION v1(3), v2(3)
264 .....
265 clasc = COS(lasc)
266 slasc = SIN(lasc)
267 caper = COS(aper)
268 saper = SIN(aper)
269 cincl = COS(incl)
270 sinc1 = SIN(incl)
271 .....
272 l1 = clasc * caper - slasc * saper * cincl
273 l2 = -clasc * saper - slasc * caper * cincl
274 l3 = slasc * cincl
275 m1 = slasc * caper + clasc * saper * cincl
276 m2 = -slasc * saper + clasc * caper * cincl
277 m3 = -clasc * sinc1
278 n1 = saper * sinc1
279 n2 = caper * sinc1
280 n3 = cincl
281 .....
282 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
283 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
284 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
285 END
286
287 .....
288 SUBROUTINE Trans3 (lasc, aper, incl, v1, v2)
289 .....
290 * This routine will transform the inertial cartesian vector, v1 to
291 * the perifocal vector, v2 given the longitude of the ascending node,
292 * argument of periastron and the inclination angle.
293 .....
294 * v2(1) = eccentricity direction
295 * v2(2) = parameter direction
296 * v2(3) = orbit normal direction
297 .....
298 * v1(1) = x direction
299 * v1(2) = y direction
300 * v1(3) = z direction
301 .....
302 IMPLICIT NONE
303 INTEGER i, j
304 REAL*8 lasc, aper, incl, r, v1, v2
305 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
306 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
307 DIMENSION v1(3), v2(3)
308 .....
309 clasc = COS(lasc)
310 slasc = SIN(lasc)
311 caper = COS(aper)
312 saper = SIN(aper)
313 cincl = COS(incl)
314 sinc1 = SIN(incl)
315 .....
316 l1 = clasc * caper - slasc * saper * cincl
317 l2 = -clasc * saper - slasc * caper * cincl
318 l3 = slasc * cincl
319 m1 = slasc * caper + clasc * saper * cincl
320 m2 = -slasc * saper + clasc * caper * cincl
321 m3 = -clasc * sinc1
322 n1 = saper * sinc1
323 n2 = caper * sinc1
324 n3 = cincl
325 .....
326 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
327 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
328 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
329 END
330
331 .....
332 SUBROUTINE Trans4 (lasc, aper, incl, v1, v2)
333 .....
334 * This routine will transform the inertial cartesian vector, v1 to
335 * the perifocal vector, v2 given the longitude of the ascending node,
336 * argument of periastron and the inclination angle.
337 .....
338 * v2(1) = eccentricity direction
339 * v2(2) = parameter direction
340 * v2(3) = orbit normal direction
341 .....
342 * v1(1) = x direction
343 * v1(2) = y direction
344 * v1(3) = z direction
345 .....
346 IMPLICIT NONE
347 INTEGER i, j
348 REAL*8 lasc, aper, incl, r, v1, v2
349 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
350 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
351 DIMENSION v1(3), v2(3)
352 .....
353 clasc = COS(lasc)
354 slasc = SIN(lasc)
355 caper = COS(aper)
356 saper = SIN(aper)
357 cincl = COS(incl)
358 sinc1 = SIN(incl)
359 .....
360 l1 = clasc * caper - slasc * saper * cincl
361 l2 = -clasc * saper - slasc * caper * cincl
362 l3 = slasc * cincl
363 m1 = slasc * caper + clasc * saper * cincl
364 m2 = -slasc * saper + clasc * caper * cincl
365 m3 = -clasc * sinc1
366 n1 = saper * sinc1
367 n2 = caper * sinc1
368 n3 = cincl
369 .....
370 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
371 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
372 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
373 END
374
375 .....
376 SUBROUTINE Trans5 (lasc, aper, incl, v1, v2)
377 .....
378 * This routine will transform the inertial cartesian vector, v1 to
379 * the perifocal vector, v2 given the longitude of the ascending node,
380 * argument of periastron and the inclination angle.
381 .....
382 * v2(1) = eccentricity direction
383 * v2(2) = parameter direction
384 * v2(3) = orbit normal direction
385 .....
386 * v1(1) = x direction
387 * v1(2) = y direction
388 * v1(3) = z direction
389 .....
390 IMPLICIT NONE
391 INTEGER i, j
392 REAL*8 lasc, aper, incl, r, v1, v2
393 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
394 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
395 DIMENSION v1(3), v2(3)
396 .....
397 clasc = COS(lasc)
398 slasc = SIN(lasc)
399 caper = COS(aper)
400 saper = SIN(aper)
401 cincl = COS(incl)
402 sinc1 = SIN(incl)
403 .....
404 l1 = clasc * caper - slasc * saper * cincl
405 l2 = -clasc * saper - slasc * caper * cincl
406 l3 = slasc * cincl
407 m1 = slasc * caper + clasc * saper * cincl
408 m2 = -slasc * saper + clasc * caper * cincl
409 m3 = -clasc * sinc1
410 n1 = saper * sinc1
411 n2 = caper * sinc1
412 n3 = cincl
413 .....
414 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
415 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
416 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
417 END
418
419 .....
420 SUBROUTINE Trans6 (lasc, aper, incl, v1, v2)
421 .....
422 * This routine will transform the inertial cartesian vector, v1 to
423 * the perifocal vector, v2 given the longitude of the ascending node,
424 * argument of periastron and the inclination angle.
425 .....
426 * v2(1) = eccentricity direction
427 * v2(2) = parameter direction
428 * v2(3) = orbit normal direction
429 .....
430 * v1(1) = x direction
431 * v1(2) = y direction
432 * v1(3) = z direction
433 .....
434 IMPLICIT NONE
435 INTEGER i, j
436 REAL*8 lasc, aper, incl, r, v1, v2
437 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
438 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
439 DIMENSION v1(3), v2(3)
440 .....
441 clasc = COS(lasc)
442 slasc = SIN(lasc)
443 caper = COS(aper)
444 saper = SIN(aper)
445 cincl = COS(incl)
446 sinc1 = SIN(incl)
447 .....
448 l1 = clasc * caper - slasc * saper * cincl
449 l2 = -clasc * saper - slasc * caper * cincl
450 l3 = slasc * cincl
451 m1 = slasc * caper + clasc * saper * cincl
452 m2 = -slasc * saper + clasc * caper * cincl
453 m3 = -clasc * sinc1
454 n1 = saper * sinc1
455 n2 = caper * sinc1
456 n3 = cincl
457 .....
458 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
459 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
460 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
461 END
462
463 .....
464 SUBROUTINE Trans7 (lasc, aper, incl, v1, v2)
465 .....
466 * This routine will transform the inertial cartesian vector, v1 to
467 * the perifocal vector, v2 given the longitude of the ascending node,
468 * argument of periastron and the inclination angle.
469 .....
470 * v2(1) = eccentricity direction
471 * v2(2) = parameter direction
472 * v2(3) = orbit normal direction
473 .....
474 * v1(1) = x direction
475 * v1(2) = y direction
476 * v1(3) = z direction
477 .....
478 IMPLICIT NONE
479 INTEGER i, j
480 REAL*8 lasc, aper, incl, r, v1, v2
481 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
482 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
483 DIMENSION v1(3), v2(3)
484 .....
485 clasc = COS(lasc)
486 slasc = SIN(lasc)
487 caper = COS(aper)
488 saper = SIN(aper)
489 cincl = COS(incl)
490 sinc1 = SIN(incl)
491 .....
492 l1 = clasc * caper - slasc * saper * cincl
493 l2 = -clasc * saper - slasc * caper * cincl
494 l3 = slasc * cincl
495 m1 = slasc * caper + clasc * saper * cincl
496 m2 = -slasc * saper + clasc * caper * cincl
497 m3 = -clasc * sinc1
498 n1 = saper * sinc1
499 n2 = caper * sinc1
500 n3 = cincl
501 .....
502 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
503 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
504 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
505 END
506
507 .....
508 SUBROUTINE Trans8 (lasc, aper, incl, v1, v2)
509 .....
510 * This routine will transform the inertial cartesian vector, v1 to
511 * the perifocal vector, v2 given the longitude of the ascending node,
512 * argument of periastron and the inclination angle.
513 .....
514 * v2(1) = eccentricity direction
515 * v2(2) = parameter direction
516 * v2(3) = orbit normal direction
517 .....
518 * v1(1) = x direction
519 * v1(2) = y direction
520 * v1(3) = z direction
521 .....
522 IMPLICIT NONE
523 INTEGER i, j
524 REAL*8 lasc, aper, incl, r, v1, v2
525 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
526 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
527 DIMENSION v1(3), v2(3)
528 .....
529 clasc = COS(lasc)
530 slasc = SIN(lasc)
531 caper = COS(aper)
532 saper = SIN(aper)
533 cincl = COS(incl)
534 sinc1 = SIN(incl)
535 .....
536 l1 = clasc * caper - slasc * saper * cincl
537 l2 = -clasc * saper - slasc * caper * cincl
538 l3 = slasc * cincl
539 m1 = slasc * caper + clasc * saper * cincl
540 m2 = -slasc * saper + clasc * caper * cincl
541 m3 = -clasc * sinc1
542 n1 = saper * sinc1
543 n2 = caper * sinc1
544 n3 = cincl
545 .....
546 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
547 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
548 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
549 END
550
551 .....
552 SUBROUTINE Trans9 (lasc, aper, incl, v1, v2)
553 .....
554 * This routine will transform the inertial cartesian vector, v1 to
555 * the perifocal vector, v2 given the longitude of the ascending node,
556 * argument of periastron and the inclination angle.
557 .....
558 * v2(1) = eccentricity direction
559 * v2(2) = parameter direction
560 * v2(3) = orbit normal direction
561 .....
562 * v1(1) = x direction
563 * v1(2) = y direction
564 * v1(3) = z direction
565 .....
566 IMPLICIT NONE
567 INTEGER i, j
568 REAL*8 lasc, aper, incl, r, v1, v2
569 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
570 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
571 DIMENSION v1(3), v2(3)
572 .....
573 clasc = COS(lasc)
574 slasc = SIN(lasc)
575 caper = COS(aper)
576 saper = SIN(aper)
577 cincl = COS(incl)
578 sinc1 = SIN(incl)
579 .....
580 l1 = clasc * caper - slasc * saper * cincl
581 l2 = -clasc * saper - slasc * caper * cincl
582 l3 = slasc * cincl
583 m1 = slasc * caper + clasc * saper * cincl
584 m2 = -slasc * saper + clasc * caper * cincl
585 m3 = -clasc * sinc1
586 n1 = saper * sinc1
587 n2 = caper * sinc1
588 n3 = cincl
589 .....
590 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
591 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
592 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
593 END
594
595 .....
596 SUBROUTINE Trans10 (lasc, aper, incl, v1, v2)
597 .....
598 * This routine will transform the inertial cartesian vector, v1 to
599 * the perifocal vector, v2 given the longitude of the ascending node,
600 * argument of periastron and the inclination angle.
601 .....
602 * v2(1) = eccentricity direction
603 * v2(2) = parameter direction
604 * v2(3) = orbit normal direction
605 .....
606 * v1(1) = x direction
607 * v1(2) = y direction
608 * v1(3) = z direction
609 .....
610 IMPLICIT NONE
611 INTEGER i, j
612 REAL*8 lasc, aper, incl, r, v1, v2
613 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
614 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
615 DIMENSION v1(3), v2(3)
616 .....
617 clasc = COS(lasc)
618 slasc = SIN(lasc)
619 caper = COS(aper)
620 saper = SIN(aper)
621 cincl = COS(incl)
622 sinc1 = SIN(incl)
623 .....
624 l1 = clasc * caper - slasc * saper * cincl
625 l2 = -clasc * saper - slasc * caper * cincl
626 l3 = slasc * cincl
627 m1 = slasc * caper + clasc * saper * cincl
628 m2 = -slasc * saper + clasc * caper * cincl
629 m3 = -clasc * sinc1
630 n1 = saper * sinc1
631 n2 = caper * sinc1
632 n3 = cincl
633 .....
634 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
635 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
636 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
637 END
638
639 .....
640 SUBROUTINE Trans11 (lasc, aper, incl, v1, v2)
641 .....
642 * This routine will transform the inertial cartesian vector, v1 to
643 * the perifocal vector, v2 given the longitude of the ascending node,
644 * argument of periastron and the inclination angle.
645 .....
646 * v2(1) = eccentricity direction
647 * v2(2) = parameter direction
648 * v2(3) = orbit normal direction
649 .....
650 * v1(1) = x direction
651 * v1(2) = y direction
652 * v1(3) = z direction
653 .....
654 IMPLICIT NONE
655 INTEGER i, j
656 REAL*8 lasc, aper, incl, r, v1, v2
657 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
658 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
659 DIMENSION v1(3), v2(3)
660 .....
661 clasc = COS(lasc)
662 slasc = SIN(lasc)
663 caper = COS(aper)
664 saper = SIN(aper)
665 cincl = COS(incl)
666 sinc1 = SIN(incl)
667 .....
668 l1 = clasc * caper - slasc * saper * cincl
669 l2 = -clasc * saper - slasc * caper * cincl
670 l3 = slasc * cincl
671 m1 = slasc * caper + clasc * saper * cincl
672 m2 = -slasc * saper + clasc * caper * cincl
673 m3 = -clasc * sinc1
674 n1 = saper * sinc1
675 n2 = caper * sinc1
676 n3 = cincl
677 .....
678 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
679 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
680 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
681 END
682
683 .....
684 SUBROUTINE Trans12 (lasc, aper, incl, v1, v2)
685 .....
686 * This routine will transform the inertial cartesian vector, v1 to
687 * the perifocal vector, v2 given the longitude of the ascending node,
688 * argument of periastron and the inclination angle.
689 .....
690 * v2(1) = eccentricity direction
691 * v2(2) = parameter direction
692 * v2(3) = orbit normal direction
693 .....
694 * v1(1) = x direction
695 * v1(2) = y direction
696 * v1(3) = z direction
697 .....
698 IMPLICIT NONE
699 INTEGER i, j
700 REAL*8 lasc, aper, incl, r, v1, v2
701 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
702 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
703 DIMENSION v1(3), v2(3)
704 .....
705 clasc = COS(lasc)
706 slasc = SIN(lasc)
707 caper = COS(aper)
708 saper = SIN(aper)
709 cincl = COS(incl)
710 sinc1 = SIN(incl)
711 .....
712 l1 = clasc * caper - slasc * saper * cincl
713 l2 = -clasc * saper - slasc * caper * cincl
714 l3 = slasc * cincl
715 m1 = slasc * caper + clasc * saper * cincl
716 m2 = -slasc * saper + clasc * caper * cincl
717 m3 = -clasc * sinc1
718 n1 = saper * sinc1
719 n2 = caper * sinc1
720 n3 = cincl
721 .....
722 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
723 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
724 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
725 END
726
727 .....
728 SUBROUTINE Trans13 (lasc, aper, incl, v1, v2)
729 .....
730 * This routine will transform the inertial cartesian vector, v1 to
731 * the perifocal vector, v2 given the longitude of the ascending node,
732 * argument of periastron and the inclination angle.
733 .....
734 * v2(1) = eccentricity direction
735 * v2(2) = parameter direction
736 * v2(3) = orbit normal direction
737 .....
738 * v1(1) = x direction
739 * v1(2) = y direction
740 * v1(3) = z direction
741 .....
742 IMPLICIT NONE
743 INTEGER i, j
744 REAL*8 lasc, aper, incl, r, v1, v2
745 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
746 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
747 DIMENSION v1(3), v2(3)
748 .....
749 clasc = COS(lasc)
750 slasc = SIN(lasc)
751 caper = COS(aper)
752 saper = SIN(aper)
753 cincl = COS(incl)
754 sinc1 = SIN(incl)
755 .....
756 l1 = clasc * caper - slasc * saper * cincl
757 l2 = -clasc * saper - slasc * caper * cincl
758 l3 = slasc * cincl
759 m1 = slasc * caper + clasc * saper * cincl
760 m2 = -slasc * saper + clasc * caper * cincl
761 m3 = -clasc * sinc1
762 n1 = saper * sinc1
763 n2 = caper * sinc1
764 n3 = cincl
765 .....
766 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
767 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
768 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
769 END
770
771 .....
772 SUBROUTINE Trans14 (lasc, aper, incl, v1, v2)
773 .....
774 * This routine will transform the inertial cartesian vector, v1 to
775 * the perifocal vector, v2 given the longitude of the ascending node,
776 * argument of periastron and the inclination angle.
777 .....
778 * v2(1) = eccentricity direction
779 * v2(2) = parameter direction
780 * v2(3) = orbit normal direction
781 .....
782 * v1(1) = x direction
783 * v1(2) = y direction
784 * v1(3) = z direction
785 .....
786 IMPLICIT NONE
787 INTEGER i, j
788 REAL*8 lasc, aper, incl, r, v1, v2
789 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
790 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
791 DIMENSION v1(3), v2(3)
792 .....
793 clasc = COS(lasc)
794 slasc = SIN(lasc)
795 caper = COS(aper)
796 saper = SIN(aper)
797 cincl = COS(incl)
798 sinc1 = SIN(incl)
799 .....
800 l1 = clasc * caper - slasc * saper * cincl
801 l2 = -clasc * saper - slasc * caper * cincl
802 l3 = slasc * cincl
803 m1 = slasc * caper + clasc * saper * cincl
804 m2 = -slasc * saper + clasc * caper * cincl
805 m3 = -clasc * sinc1
806 n1 = saper * sinc1
807 n2 = caper * sinc1
808 n3 = cincl
809 .....
810 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
811 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
812 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
813 END
814
815 .....
816 SUBROUTINE Trans15 (lasc, aper, incl, v1, v2)
817 .....
818 * This routine will transform the inertial cartesian vector, v1 to
819 * the perifocal vector, v2 given the longitude of the ascending node,
820 * argument of periastron and the inclination angle.
821 .....
822 * v2(1) = eccentricity direction
823 * v2(2) = parameter direction
824 * v2(3) = orbit normal direction
825 .....
826 * v1(1) = x direction
827 * v1(2) = y direction
828 * v1(3) = z direction
829 .....
830 IMPLICIT NONE
831 INTEGER i, j
832 REAL*8 lasc, aper, incl, r, v1, v2
833 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
834 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
835 DIMENSION v1(3), v2(3)
836 .....
837 clasc = COS(lasc)
838 slasc = SIN(lasc)
839 caper = COS(aper)
840 saper = SIN(aper)
841 cincl = COS(incl)
842 sinc1 = SIN(incl)
843 .....
844 l1 = clasc * caper - slasc * saper * cincl
845 l2 = -clasc * saper - slasc * caper * cincl
846 l3 = slasc * cincl
847 m1 = slasc * caper + clasc * saper * cincl
848 m2 = -slasc * saper + clasc * caper * cincl
849 m3 = -clasc * sinc1
850 n1 = saper * sinc1
851 n2 = caper * sinc1
852 n3 = cincl
853 .....
854 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
855 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
856 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
857 END
858
859 .....
860 SUBROUTINE Trans16 (lasc, aper, incl, v1, v2)
861 .....
862 * This routine will transform the inertial cartesian vector, v1 to
863 * the perifocal vector, v2 given the longitude of the ascending node,
864 * argument of periastron and the inclination angle.
865 .....
866 * v2(1) = eccentricity direction
867 * v2(2) = parameter direction
868 * v2(3) = orbit normal direction
869 .....
870 * v1(1) = x direction
871 * v1(2) = y direction
872 * v1(3) = z direction
873 .....
874 IMPLICIT NONE
875 INTEGER i, j
876 REAL*8 lasc, aper, incl, r, v1, v2
877 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
878 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
879 DIMENSION v1(3), v2(3)
880 .....
881 clasc = COS(lasc)
882 slasc = SIN(lasc)
883 caper = COS(aper)
884 saper = SIN(aper)
885 cincl = COS(incl)
886 sinc1 = SIN(incl)
887 .....
888 l1 = clasc * caper - slasc * saper * cincl
889 l2 = -clasc * saper - slasc * caper * cincl
890 l3 = slasc * cincl
891 m1 = slasc * caper + clasc * saper * cincl
892 m2 = -slasc * saper + clasc * caper * cincl
893 m3 = -clasc * sinc1
894 n1 = saper * sinc1
895 n2 = caper * sinc1
896 n3 = cincl
897 .....
898 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
899 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
900 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
901 END
902
903 .....
904 SUBROUTINE Trans17 (lasc, aper, incl, v1, v2)
905 .....
906 * This routine will transform the inertial cartesian vector, v1 to
907 * the perifocal vector, v2 given the longitude of the ascending node,
908 * argument of periastron and the inclination angle.
909 .....
910 * v2(1) = eccentricity direction
911 * v2(2) = parameter direction
912 * v2(3) = orbit normal direction
913 .....
914 * v1(1) = x direction
915 * v1(2) = y direction
916 * v1(3) = z direction
917 .....
918 IMPLICIT NONE
919 INTEGER i, j
920 REAL*8 lasc, aper, incl, r, v1, v2
921 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
922 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
923 DIMENSION v1(3), v2(3)
924 .....
925 clasc = COS(lasc)
926 slasc = SIN(lasc)
927 caper = COS(aper)
928 saper = SIN(aper)
929 cincl = COS(incl)
930 sinc1 = SIN(incl)
931 .....
932 l1 = clasc * caper - slasc * saper * cincl
933 l2 = -clasc * saper - slasc * caper * cincl
934 l3 = slasc * cincl
935 m1 = slasc * caper + clasc * saper * cincl
936 m2 = -slasc * saper + clasc * caper * cincl
937 m3 = -clasc * sinc1
938 n1 = saper * sinc1
939 n2 = caper * sinc1
940 n3 = cincl
941 .....
942 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
943 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
944 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
945 END
946
947 .....
948 SUBROUTINE Trans18 (lasc, aper, incl, v1, v2)
949 .....
950 * This routine will transform the inertial cartesian vector, v1 to
951 * the perifocal vector, v2 given the longitude of the ascending node,
952 * argument of periastron and the inclination angle.
953 .....
954 * v2(1) = eccentricity direction
955 * v2(2) = parameter direction
956 * v2(3) = orbit normal direction
957 .....
958 * v1(1) = x direction
959 * v1(2) = y direction
960 * v1(3) = z direction
961 .....
962 IMPLICIT NONE
963 INTEGER i, j
964 REAL*8 lasc, aper, incl, r, v1, v2
965 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
966 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
967 DIMENSION v1(3), v2(3)
968 .....
969 clasc = COS(lasc)
970 slasc = SIN(lasc)
971 caper = COS(aper)
972 saper = SIN(aper)
973 cincl = COS(incl)
974 sinc1 = SIN(incl)
975 .....
976 l1 = clasc * caper - slasc * saper * cincl
977 l2 = -clasc * saper - slasc * caper * cincl
978 l3 = slasc * cincl
979 m1 = slasc * caper + clasc * saper * cincl
980 m2 = -slasc * saper + clasc * caper * cincl
981 m3 = -clasc * sinc1
982 n1 = saper * sinc1
983 n2 = caper * sinc1
984 n3 = cincl
985 .....
986 v2(1) = l1 * v1(1) + l2 * v1(2) + l3 * v1(3)
987 v2(2) = m1 * v1(1) + m2 * v1(2) + m3 * v1(3)
988 v2(3) = n1 * v1(1) + n2 * v1(2) + n3 * v1(3)
989 END
990
991 .....
992 SUBROUTINE Trans19 (lasc, aper, incl, v1, v2)
993 .....
994 * This routine will transform the inertial cartesian vector, v1 to
995 * the perifocal vector, v2 given the longitude of the ascending node,
996 * argument of periastron and the inclination angle.
997 .....
998 * v2(1) = eccentricity direction
999 * v2(2) = parameter direction
1000 * v2(3) = orbit normal direction
1001 .....
1002 * v1(1) = x direction
1003 * v1(2) = y direction
1004 * v1(3) = z direction
1005 .....
1006 IMPLICIT NONE
1007 INTEGER i, j
1008 REAL*8 lasc, aper, incl, r, v1, v2
1009 REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
1010 REAL*8 clasc, slasc, caper, saper, cincl, sinc1
1011 DIMENSION v1(3), v2(
```

```

277 * .....
278 * IMPLICIT NONE
279 *
280 * INTEGER i, j
281 * REAL*8 lasec, aper, incl, r, v1, v2
282 * REAL*8 l1, l2, l3, m1, m2, m3, n1, n2, n3
283 * REAL*8 clasc, slasc, caper, saper, cincl, sincl
284 * DIMENSION v1(3), v2(3)
285 * .....
286 *
287 *      clasc = COS(lasec)
288 *      slasc = SIN(lasec)
289 *      caper = COS(aper)
290 *      saper = SIN(aper)
291 *      cincl = COS(incl)
292 *      sincl = SIN(incl)
293 *
294 *      l1 = clasc * caper - slasc * saper * cincl
295 *      l2 = -clasc * saper - slasc * caper * cincl
296 *      l3 = slasc * sincl
297 *      m1 = slasc * caper + clasc * saper * cincl
298 *      m2 = -slasc * saper + clasc * caper * cincl
299 *      m3 = -clasc * sincl
300 *      n1 = saper * sincl
301 *      n2 = caper * sincl
302 *      n3 = cincl
303 *
304 *      v2(1) = l1 * v1(1) + m1 * v1(2) + n1 * v1(3)
305 *      v2(2) = l2 * v1(1) + m2 * v1(2) + n2 * v1(3)
306 *      v2(3) = l3 * v1(1) + m3 * v1(2) + n3 * v1(3)
307 *
308 *      END
309 *
310 * .....
311 * SUBROUTINE Trans3 (true, v1, v2)
312 * .....
313 *
314 * * This routine transforms polar vector, v1 to perifocal vector, v2
315 * * given the true anomaly.
316 *
317 * * v1(1) = radial direction
318 * * v1(2) = polar angle direction
319 * * v1(3) = orbit normal direction
320 *
321 * * v2(1) = eccentricity direction
322 * * v2(2) = parameter direction
323 * * v2(3) = orbit normal direction
324 *
325 * .....
326 * IMPLICIT NONE
327 * REAL*8 ctrue, strue, true, v1, v2
328 * DIMENSION v1(3), v2(3)
329 * .....
330 *      ctrue = COS(true)
331 *      strue = SIN(true)
332 *
333 *      v2(1) = ctrue * v1(1) - strue * v1(2)
334 *      v2(2) = strue * v1(1) + ctrue * v1(2)
335 *      v2(3) = v1(3)
336 *
337 *      END
338 *
339 * .....
340 * SUBROUTINE Trans4 (true, v1, v2)
341 * .....
342 *
343 * * This routine transforms perifocal vector, v1 to polar vector, v2
344 * * given the true anomaly.
345 *

```

```

346 *
347 * * v2(1) = radial direction
348 * * v2(2) = polar angle direction
349 * * v2(3) = orbit normal direction
350 *
351 * * v1(1) = eccentricity direction
352 * * v1(2) = parameter direction
353 * * v1(3) = orbit normal direction
354 *
355 * .....
356 * IMPLICIT NONE
357 * REAL*8 ctrue, strue, true, v1, v2
358 * DIMENSION v1(3), v2(3)
359 * .....
360 *      ctrue = COS(true)
361 *      strue = SIN(true)
362 *
363 *      v2(1) = ctrue * v1(1) + strue * v1(2)
364 *      v2(2) = -strue * v1(1) + ctrue * v1(2)
365 *      v2(3) = v1(3)
366 *
367 *      END
368 *
369 * .....
370 * SUBROUTINE Trans5 (v1, r, coel, ra)
371 * .....
372 *
373 * * This routine will find the radius, coelevation, right ascension given
374 * * the cartesian vector, v1.
375 *
376 * * v1(1) = x direction
377 * * v1(2) = y direction
378 * * v1(3) = z direction
379 *
380 * .....
381 * IMPLICIT NONE
382 * REAL*8 coel, ra, r, rho, v1, x, y, z
383 * DIMENSION v1(3)
384 * .....
385 *      x = v1(1)
386 *      y = v1(2)
387 *      z = v1(3)
388 *      r = SQRT(x**2 + y**2 + z**2)
389 *      rho = SQRT(x**2 + y**2)
390 *
391 *      coel = ATAN2(rho, z)
392 *      ra = ATAN2(y, x)
393 *
394 *      END
395 *
396 * .....
397 * SUBROUTINE Trans6 (r, coel, ra, v1)
398 * .....
399 *
400 * * This routine will find the cartesian vector v1 given the radius,
401 * * coelevation, and right ascension.
402 *
403 * * v1(1) = x direction
404 * * v1(2) = y direction
405 * * v1(3) = z direction
406 *
407 * .....
408 * IMPLICIT NONE
409 * REAL*8 coel, ra, r, v1, scoel
410 * DIMENSION v1(3)
411 * .....
412 *      scoel = SIN(coel)
413 *
414 *

```

```

415      v1(1) = r * scoel * COS(ra)
416      v1(2) = r * scoel * SIN(ra)
417      v1(3) = r * COS(coel)
418      .
419      END
420
421
422
423      SUBROUTINE Trans7 (v1, coel, ra, v2)
424      .....
425      .
426      . This routine will convert the spherical local tangent vector v1
427      . to inertial cartesian vector v2 given the coelevation and
428      . right ascension.
429      .
430      . v1(1) = radial direction
431      . v1(2) = coelevation direction (South)
432      . v1(3) = right ascension direction (East)
433      .
434      . v2(1) = x direction
435      . v2(2) = y direction
436      . v2(3) = z direction
437      .
438      .....
439      IMPLICIT NONE
440      REAL*8 v1, coel, ra, v2, ccoel, scoel, cra, sra
441      DIMENSION v1(3), v2(3)
442      .....
443      ccoel = COS(coel)
444      scoel = SIN(coel)
445      cra = COS(ra)
446      sra = SIN(ra)
447      .
448      v2(1) = scoel * cra * v1(1) + ccoel * cra * v1(2) - sra * v1(3)
449      v2(2) = scoel * sra * v1(1) + ccoel * sra * v1(2) + cra * v1(3)
450      v2(3) = ccoel * v1(1) - scoel * v1(2)
451      .
452      END

```



```
139 REAL*8 roll, pitch, w, rroll, rpitch, ryaw
140 REAL*8 cp, sp, tr, sr
141 DIMENSION w(3)
142 .....
143 cp = COS(pitch)
144 sp = SIN(pitch)
145 tr = TAN(roll)
146 sr = 1.80 / COS(roll)
147
148 rroll = cp * w(1) + sp * w(2) + sr * w(3)
149 rpitch = sp * tr * w(1) + w(2) - cp * tr * w(3)
150 ryaw = -sp * sr * w(1) + cp * sr * w(3)
151
152 END
153
154
155
156
157
158
159
160
161
162
163 SUBROUTINE Rate2 (roll, pitch, rroll, rpitch, ryaw, w)
164 .....
165 * This routine will convert the roll, pitch, yaw rates to cartesian
166 * Body rates w (wx, wy, wz) for 3-1-2 transformation.
167 .....
168
169 IMPLICIT NONE
170 REAL*8 roll, pitch, w, rroll, rpitch, ryaw
171 REAL*8 cp, sp, cr, sr
172 DIMENSION w(3)
173 .....
174 cp = COS(pitch)
175 sp = SIN(pitch)
176 cr = COS(roll)
177 sr = SIN(roll)
178
179 w(1) = cp * rroll - sp * cr * ryaw
180 w(2) = rpitch + sr * cr * ryaw
181 w(3) = sp * rroll + cp * cr * ryaw
182
183 END
184
```

```

1  SUBROUTINE Rkqc (y, dydx, n, x, htry, eps, yscal, hdid,
2  hnext, Derivs)
3  .....
4  IMPLICIT REAL*8 (a-h, o-z)
5  PARAMETER (nmax = 20, pgrow = -0.2d0, pshrnk = -0.25d0)
6  PARAMETER (fcor = 1.d0 / 15.d0, safety = 0.9d0, errcon = 6.d-4)
7  PARAMETER (tiny = 1.d-15)
8  EXTERNAL Derivs
9  DIMENSION y(n), dydx(n), yscal(n), ytemp(nmax), ysav(nmax)
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 .....
22 .....
23 .....
24 .....
25 .....
26 .....
27 .....
28 .....
29 .....
30 .....
31 .....
32 .....
33 .....
34 .....
35 .....
36 .....
37 .....
38 .....
39 .....
40 .....
41 .....
42 .....
43 .....
44 .....
45 .....
46 .....
47 .....
48 .....
49 .....
50 .....
51 .....
52 .....
53 .....
54 .....
55 .....
56 .....
57 .....
58 .....
59 .....
60 .....
61 .....
62 .....
63 .....
64 .....
65 .....
66 .....
67 .....
68 .....
69 .....

```

```

70 .....
71 SUBROUTINE Rk4dub(v, nvar, x1, x2, nstep, Derivs, Out, ip)
72 .....
73 IMPLICIT NONE
74 INTEGER ip, k, nmax, nstep, nvar
75 REAL*8 h, v, dv, x, x1, x2
76 PARAMETER (nmax = 10)
77 DIMENSION v(nvar), dv(nmax)
78 EXTERNAL Derivs, Out
79 .....
80 .....
81 x = x1
82 h = (x2 - x1) / nstep
83 h = SIGN(h, h)
84 IF (ip .GT. 0) THEN
85   CALL Out (v, x, h)
86   END IF
87 .....
88 .....
89 .....
90 DO k = 1, nstep
91   CALL Derivs (v, dv, x, h)
92   CALL Rk4 (v, dv, nvar, x, h, v, Derivs)
93   x = x + h
94   IF (ip .GT. 0) THEN
95     CALL Out (v, x, h)
96   END IF
97   END DO
98   END
99 .....
100 .....
101 .....
102 .....
103 .....
104 .....
105 .....
106 IMPLICIT NONE
107 INTEGER i, n, nmax
108 REAL*8 dydx, dym, dyt, h, hh, h6, y, yt, yout, x, xh
109 PARAMETER (nmax = 20)
110 DIMENSION y(n), dydx(n), yout(n), yt(nmax), dyt(nmax)
111 .....
112 .....
113 .....
114 hh = h / 2.d0
115 h6 = h / 6.d0
116 .....
117 .....
118 .....
119 xh = x + hh
120 DO i = 1, n
121   yt(i) = y(i) + hh * dydx(i)
122   END DO
123 .....
124 .....
125 .....
126 .....
127 .....
128 yt(i) = y(i) + hh * dyt(i)
129   END DO
130 .....
131 .....
132 .....
133 .....
134 .....
135 yt(i) = y(i) + h * dym(i)
136   dym(i) = dyt(i) + dym(i)
137   END DO
138 .....

```

```
139 ** fourth step
140 *
141 CALL Derivs(yt, dyt, x + h, h)
142 DO i = 1, n
143     yout(i) = y(i) + h6 * (dydx(i) + dyt(i) + 2.d0 * dym(i))
144 END DO
145 *
146 END
147
148
149
150
151
```

```

1  SUBROUTINE Gravgr (w0, tgg)
2  .....
3  * This routine will calculate the gravity gradient torque for a
4  * circular orbit
5  .....
6  * IMPLICIT NONE
7  .....
8  * REAL*8 a, ix, iy, iz, temp, tgg, w0
9  .....
10 * DIMENSION a(3, 3), tgg(3)
11 .....
12 * COMMON /inertia/ ix, iy, iz
13 * COMMON /cosmat/ a
14 .....
15 * temp = 3.d0 * w0 * w0
16 .....
17 * tgg(1) = (ix - iy) * a(2, 3) * a(3, 3) * temp
18 * tgg(2) = (ix - iz) * a(1, 3) * a(3, 3) * temp
19 * tgg(3) = (iy - iz) * a(1, 3) * a(2, 3) * temp
20 .....
21 * END
22 .....
23 .....
24 .....
25 .....
26 .....
27 .....
28 .....
29 .....
30 * This routine will calculate the torque generated by magnetic
31 * torque rods using a bang-bang control law.
32 .....
33 * IMPLICIT NONE
34 .....
35 * INTEGER i, j, k, kmax
36 .....
37 * REAL*8 ix, iy, iz, msat, scale, gp, gd
38 * REAL*8 a, roll, pitch, yaw
39 * REAL*8 g, h, b, bmag, bold, bdot, bi
40 * REAL*8 mu, semi, ecc, inclin, lasec, aper, tranom, motion, epoch
41 * REAL*8 mean0, period, r, coelv, ra, long, green0, green, earate
42 * REAL*8 r1, v1
43 * REAL*8 rmax, pmax, ymax
44 * REAL*8 m, tm, t, oldt, delt, y
45 * REAL*8 twopi
46 .....
47 * PARAMETER (kmax = 8, earate = 7.292116d-05)
48 * PARAMETER (twopi = 6.283185308d0)
49 .....
50 * DIMENSION a(3, 3), b(3), g(0:kmax, 0:kmax), h(0:kmax, 0:kmax)
51 * DIMENSION m(3), tm(3), y(7), msat(3), scale(3)
52 * DIMENSION gp(3), gd(3), bdot(3), bold(3), r1(3), v1(3), bi(3)
53 .....
54 * COMMON /inertia/ ix, iy, iz
55 * COMMON /cosmat/ a
56 * COMMON /gains/ gp, gd
57 * COMMON /geom/ g, h, k
58 * COMMON /orbit/ mu, semi, ecc, inclin, lasec, aper, tranom,
59 * motion, mean0, period
60 * COMMON /time/ epoch, green0, green
61 * COMMON /rods/ msat, scale
62 * COMMON /attmax/ rmax, pmax, ymax
63 * COMMON /control/ m
64 * COMMON /magn/ b, bmag, bdot
65 * COMMON /oldb/ bold, oldt
66 * COMMON /posvel/ r1, v1
67 .....
68 .....
69 .....

```

```

70 * calculate geomagnetic field vector
71 .....
72 * CALL Trans5 (r1, r, coelv, ra)
73 * green = green0 + earate * t
74 * green = MOD(MOD(green + twopi), twopi), twopi)
75 * long = ra - green
76 * long = MOD(MOD(long + twopi), twopi), twopi)
77 * CALL Magnet (g, h, k, long, coelv, r, green, b, bmag)
78 * CALL Trans7 (b, coelv, ra, bi)
79 .....
80 * convert to spacecraft reference frame
81 .....
82 * CALL Trans2 (lasec, aper, inclin, bi, b)
83 * CALL Trans4 (tranom, b, bi)
84 * bi(1) = -bi(1)
85 * bi(3) = -bi(3)
86 .....
87 * convert to spacecraft body coordinates
88 .....
89 * CALL Body (a, bi, b)
90 .....
91 * calculate b-dot
92 .....
93 * delt = t - oldt
94 * IF (ABS(delt) .LT. 1.d-14) THEN
95 *   tm(1) = 0.d0
96 *   tm(2) = 0.d0
97 *   tm(3) = 0.d0
98 *   RETURN
99 * END IF
100 * bdot(1) = (b(1) - bold(1)) / delt
101 * bdot(2) = (b(2) - bold(2)) / delt
102 * bdot(3) = (b(3) - bold(3)) / delt
103 .....
104 * calculate commanded dipole
105 .....
106 * IF (bdot(1) .LT. 0.d0) THEN
107 *   m(1) = msat(1) * gp(1)
108 * ELSE
109 *   IF (bdot(1) .GT. 0.d0) THEN
110 *     m(1) = -msat(1) * gp(1)
111 *   ELSE
112 *     m(1) = 0.d0
113 *   END IF
114 * END IF
115 .....
116 * IF (bdot(2) .LT. 0.d0) THEN
117 *   m(2) = msat(2) * gp(2)
118 * ELSE
119 *   IF (bdot(2) .GT. 0.d0) THEN
120 *     m(2) = -msat(2) * gp(2)
121 *   ELSE
122 *     m(2) = 0.d0
123 *   END IF
124 * END IF
125 .....
126 * IF (bdot(3) .LT. 0.d0) THEN
127 *   m(3) = msat(3) * gp(3)
128 * ELSE
129 *   IF (bdot(3) .GT. 0.d0) THEN
130 *     m(3) = -msat(3) * gp(3)
131 *   ELSE
132 *     m(3) = 0.d0
133 *   END IF
134 * END IF
135 .....
136 * calculate torque
137 .....
138 * tm(1) = m(2) * b(3) - m(3) * b(2)

```

May 31 1993 19:57:53		torques.f	Page 3
139		$tm(2) = m(3) * b(1) - m(1) * b(3)$	
140	.	$tm(3) = m(1) * b(2) - m(2) * b(1)$	
141			
142		END	
143			
144			
145			

Appendix C

Reference Papers

Appendix C.2

Structures

DESIGN, FABRICATION, AND TESTING OF A GRAPHITE-EPOXY COMPOSITE GRAVITY-GRADIENT BOOM FOR A SMALL SATELLITE

Darby G. Cooper¹ and Sean R. Olin²
 Dr. T. J. McDaniel³ and Dr. Leverage K. Seversike⁴, Advisors
 Iowa State University
 Ames, Iowa 50011

Abstract

Passive stabilization methods for satellites have undergone extensive research and development. Recently the number of small satellites (satellites less than 100 kg.) has increased dramatically. This has lead to increased use of passive stabilization methods, such as gravity-gradient. The core of a gravity-gradient stabilization system is a deployable boom with a damping mechanism. Traditionally, this boom is constructed from metal alloys. Uneven heating and cooling occurs when these alloys are exposed to varying solar radiation conditions. This can induce thermal vibrations which can lead to undesired satellite attitude inversions. Graphite-epoxy composites can be fabricated to minimize thermal expansion. This will be beneficial when applied to gravity-gradient booms. The goal of this project is to demonstrate the use of graphite-epoxy composites in gravity-gradient booms. This project encompasses: the use of a satellite attitude simulation program for boom sizing and determination of gravity-gradient boom loading, development of joint-locking mechanisms for boom deployment, and selection and testing of appropriate fabrication methods.

Gravity-Gradient Stability

The Earth's gravitational field provides a ready source of stability for satellites in low Earth orbit. The inverse square nature of the Earth's gravitational field causes a torque to act on a spacecraft in orbit. This torque causes the spacecraft's principal axis of minimum moment of inertia to align itself with the local vertical.¹

The magnitude of the torque is a function of the distance from the center of the earth (orbital radius) and the ratios of the mass moments of inertia. A cartesian coordinate system is introduced with the origin at the spacecraft center of mass, with the x-axis along the orbital velocity vector, y-axis perpendicular to the orbital plane, and the z-axis nadir pointing and completing a right-handed system.

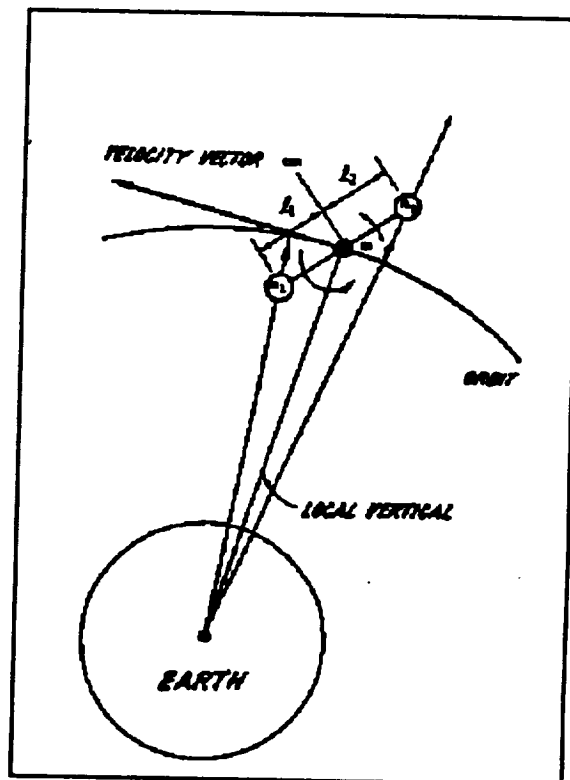


Figure 1. Spacecraft Axis System²

¹ Student, Aerospace Engineering
 Member AIAA
² Student, Aerospace Engineering
 Member AIAA

³ Professor, Aerospace Engineering
 Member AIAA
⁴ Associate Professor, Aerospace Engineering
 Member AIAA

Figure 1 shows a model satellite in Earth orbit for reference. The spacecraft's stability in a gravity-field can be measured by the introduction of three stability criteria³:

$$\Theta_x = (I_y - I_z) / I_x \quad (1a)$$

$$\Theta_y = (I_z - I_x) / I_y \quad (1b)$$

$$\Theta_z = (I_x - I_y) / I_z \quad (1c)$$

These criteria represent the stability of a satellite as the ratios of mass moments vary. This is shown graphically in Figure 2. Most spacecraft are symmetric about the z-axis so that $I_x = I_y$. When $I_x = I_y$, the region of stability lies along the Θ_x axis, and the only criterion of interest becomes Θ_x . With an axisymmetric spacecraft, equation 1a above reduces to:

$$\Theta_x = 1 - (I_z / I_x) \quad (2)$$

The determining factor for stability then becomes the ratio of I_z to I_x . Note that as I_z increases, Θ_x approaches its limiting value of unity. Equation 2 suggests that shapes with an I_z/I_x ratio near zero are the most stable. This corresponds geometrically to long, slender objects.

Spacecraft design considerations such as packing arrangements, thermal control, and available launch envelopes do not always allow for the use of long, slender bodies. However, the mass moments of inertia of spacecraft may be modified once in orbit to achieve the desired stability. While many possibilities exist to achieve a favorable mass moment ratio, (I_z/I_x) one of the most common methods is the deployment of a long boom with a tip mass; more often known as a gravity-gradient boom.

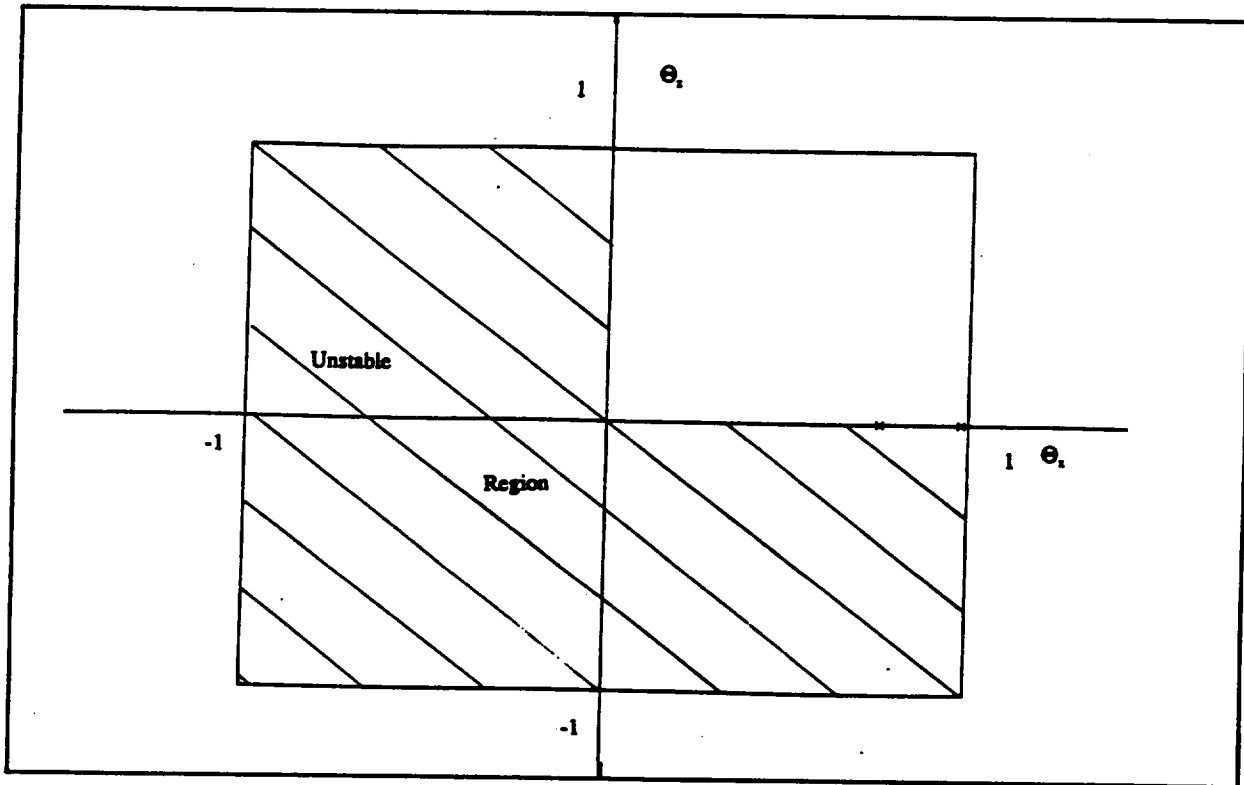


Figure 2. Stability Field

Gravity-Gradient Booms

Considerable research effort has been directed toward development and testing of gravity-gradient booms. These efforts include the development of several types of booms. A bi-metallic boom was most frequently used. This consists of two thin strips of metallic alloy approximately 0.005 centimeters. (0.002 in.) thick. Beryllium-aluminum and beryllium-copper were commonly used in this application. These two strips were carried into orbit in coils, much as common tape measures are stored. A motorized unit would unwind the two tapes. The tapes would then buckle together, inter-locking to form a closed section. This type of boom continues to be used today. Figure 3 shows some representative of this type of boom.

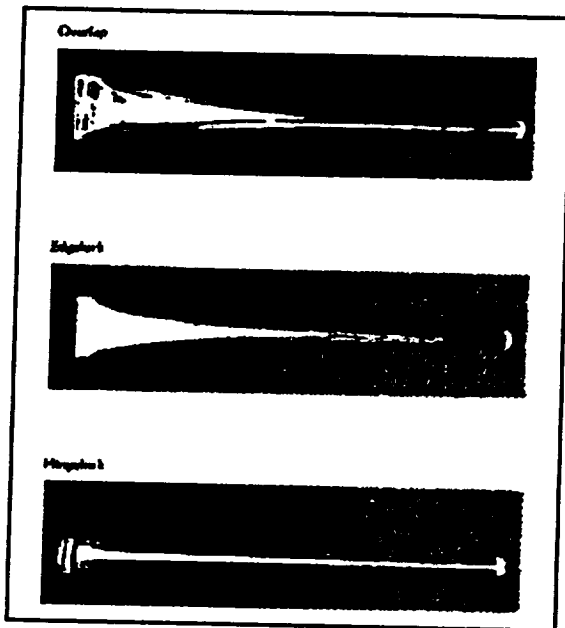


Figure 3. Tape boom examples²

Several difficulties arose with this design. Despite the closed section, the deployed boom is weak in torsion. The design has also been observed to experience thermal flutter. Thermal flutter is the bending of the boom out of the desired plane due to uneven heating. This bending changes the spacecraft's inertia properties and causes undesirable attitude behavior. The behavior has been severe enough to cause satellite attitude inversions as

documented in the HILAT and Polar BEAR^{4, 5} spacecraft.

As more small satellites are constructed, complexity becomes a planning consideration. The support system for the tape-design booms includes a motorized deployment system. This system introduces mechanical uncertainty and power requirements into the design of a spacecraft. Simplification of this system would reduce spacecraft complexity and remove potential sources of failure.

Composite Material Application

The two most significant drawbacks of the currently available gravity-gradient booms are their complexity and undesirable thermal behavior. Only limited success has been achieved with the use of coatings to control the thermal properties of metal alloys. The mechanisms for deployment of metal alloy booms also remain complex. The application of composite material technology can be used to address these two considerations.

Composite materials can be fabricated with favorable thermal expansion characteristics. By varying ply orientations and sequences, it is possible to create a material with a near zero coefficient of thermal expansion. Composite materials designed with low thermal expansion coefficients could be applied to current design configurations. This should reduce the effects of thermal flutter and its associated satellite attitude disturbances.

The mechanical properties of composite materials also allow for several possible design changes. The favorable strength to weight ratio of composite materials allows for smaller closed section designs. This facilitates the use of a telescoping design which can be carried internally as a closed section. Telescoping closed section designs do not require a drive motor for deployment. A closed section may be deployed by a spring system or by use of a gas charge to pressurize the internal volume of the boom. Either of these deployment systems would require less volume and would be less complex than a motorized unit.

Boom Section Development

Work began to study various boom section configurations to determine their feasibility. At this point, no consideration was given to final boom size (length, diameter, tip mass, etc.) or loading conditions. Rather, efforts were focused on the cross-sectional properties and methods of deployment.

Figure 4 shows the first open-section concept. This design would consist of two "C" sections with flanges attached. To store this structure, the smaller radius "C" section would have to buckle into the larger "C" section.

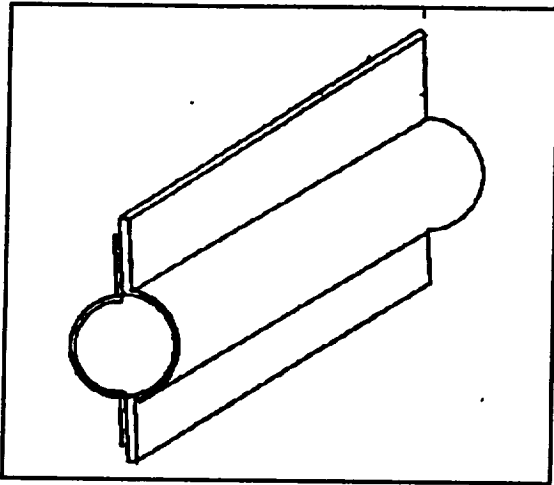


Figure 4. Open-Section Concept 1

Two test specimens of this design were fabricated. The first test specimen consisted of three plies of uniaxial graphite-epoxy composite $[0^\circ, 90^\circ, 0^\circ]$. The specimen was oven-cured under vacuum. Two "C" sections were fabricated in succession, with the first, smaller section serving as a mold for the second, larger section. These two portions were then epoxied together to form a closed section. This specimen was evaluated qualitatively to determine its buckling ability. It was found that the smaller section would not buckle without loss of matrix integrity.

The second specimen consisted of two plies of $[0, 90]$ woven graphite-epoxy composite. This specimen was constructed using the same method as the first and was also qualitatively evaluated. It was found that

the woven material was still too "stiff" to buckle without significant cracks developing in the composite matrix.

Figure 5 shows the second open-section concept. This design would consist of concentric, interlocking "C" sections. The inner, smaller radius section, would slide in rails provided by the outer sections. This design would provide for a telescoping boom.

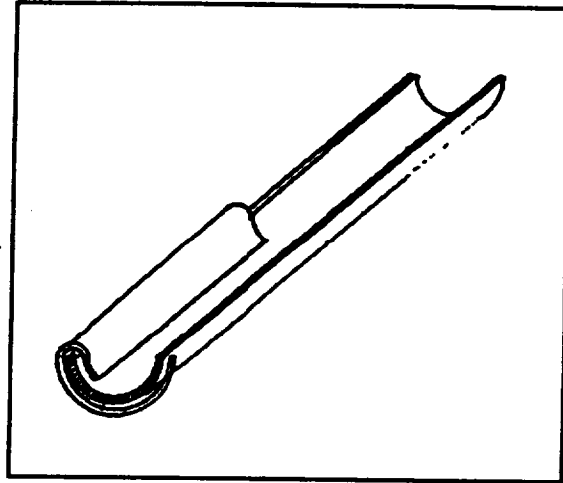


Figure 5. Open-Section Concept 2

Two test specimens of this design were fabricated. The first, or inner section, was fabricated as a "C" section and cured in an oven. The second, or outer section, which provides the slide rails, was then formed over the inner section and cured. After the second cure, it was discovered that the two sections had fused. The fabrication process used did not adequately separate the two sections to prevent epoxy from joining the two sections during the second cure. It was not possible to evaluate this specimen. The second specimen was constructed in the same manner, but with added release material to prevent the fusing of the inner and outer "C" sections. This specimen also used a third outer section. This specimen was again qualitatively evaluated. It proved difficult to fabricate a section with uniform guide rails. This created points along the structure that would bind during deployment or retraction. This difficulty arises out of the fabrication technique and the mandrel that was used. It is possible that with a different mandrel this problem could be corrected. In addition to deployment and retraction problems, this section proved very weak in torsion because of the open-

section design. The open-section also exhibits non-uniform bending stiffness, as the test specimen was found to bend very easily in the open direction.

Figure 6 shows the closed-section concept. This design consists of concentric circular sections. This concept would be used much like a telescoping radio antenna with the concentric sections being slid along one another.

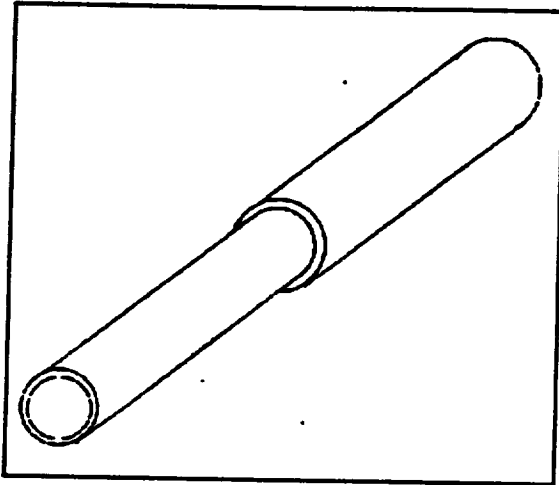


Figure 6. Closed-Section Concept

Deployment/Retraction Methods

Deployment schemes were considered in the development of the three concepts outlined above. The double "C" section (Figure 4) was considered to be deployable and retractable by buckling of the smaller radius "C" section into the larger section, and then "rolling" these two sections on a spool (much like a retractable tape measure). The interlocking "C" section (Figure 5) would be deployed and retracted by telescoping, as would the closed-section.

Several mechanisms were considered to perform these tasks. A servomechanism would be required to deploy and retract the double "C" section by rotating a spool. This would add weight and complexity to the system. A boom of this type would be locked into place by the un-buckling of the smaller section.

The interlocking "C" section could be deployed by a spring. Retraction of this design would be more

difficult. One possibility was to attach a cord to the outer section and then reel this cord in to retract the boom. It is still uncertain what type of locking mechanism could be employed to lock the boom in a deployed configuration.

The closed-section could be deployed by a gas charge. This was the simplest mechanism considered. The use of a cord and reel system could provide retraction capability. It may also be possible to design the magnetic portion of the satellite control system to avoid the need to retract the boom once it is deployed.

Boom Section Selection

The closed-section design was selected for further development. This section was chosen for its torsional properties and its uniform bending stiffness. The simplicity of the deployment mechanism was also a consideration. The current assumption is that the magnetic portion of the satellite attitude control system will be able to invert the satellite with the boom deployed, so the boom will only be deployed once.

Figure 7 and Figure 8 show the details of the joint design. Figure 7 is a cut-away of the outer wall of a section. Figure 8 is a view of an inner section, with the outer section not shown. This design provides for a forward stop collar located on the inner wall of the outer boom section, and a series of forward segmented stops located on the outer wall of the inner boom section. A segmented stop collar is also located on the inner wall of the outer section. A rear stop collar is also located on the inner section. When pressurized, the boom extends until the forward stop collar on the outer section comes into contact with the segmented collar on the inner section. The segments on the inner section slide through the gaps in the collar on the outer section, and are then rotated to lock the boom in its deployed state. The joint design also provides for the placement of three gaskets: one each at the base of the inner section, the top and base of the rear stop collar to provide a gas seal for the pressurized deployment of the boom.

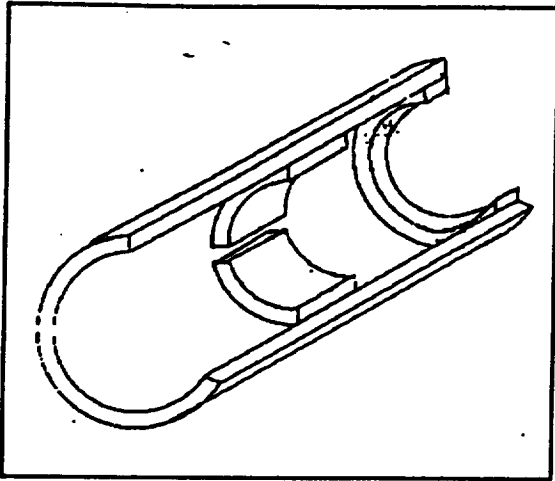


Figure 7. Joint Detail A

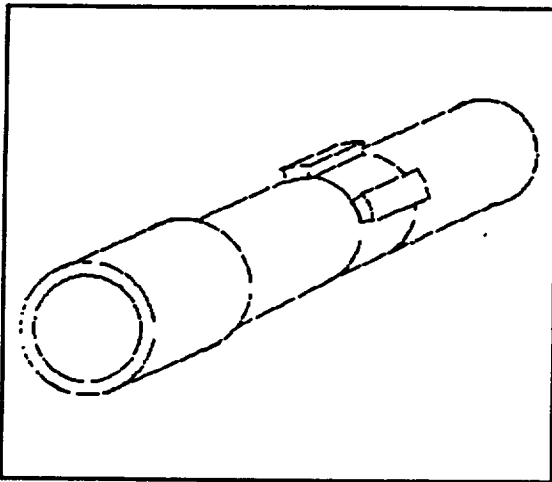


Figure 8. Joint Detail B

Boom Sizing

The operational task of a gravity-gradient boom is to modify the inertia properties of a spacecraft to increase its stability. The degree of stability increase desired is the driving factor in sizing the boom's length and tip mass to a particular satellite. Iowa State University is currently undertaking an effort to design, construct and launch a small satellite, ISAT-1 (Iowa Satellite One). The preliminary design of this

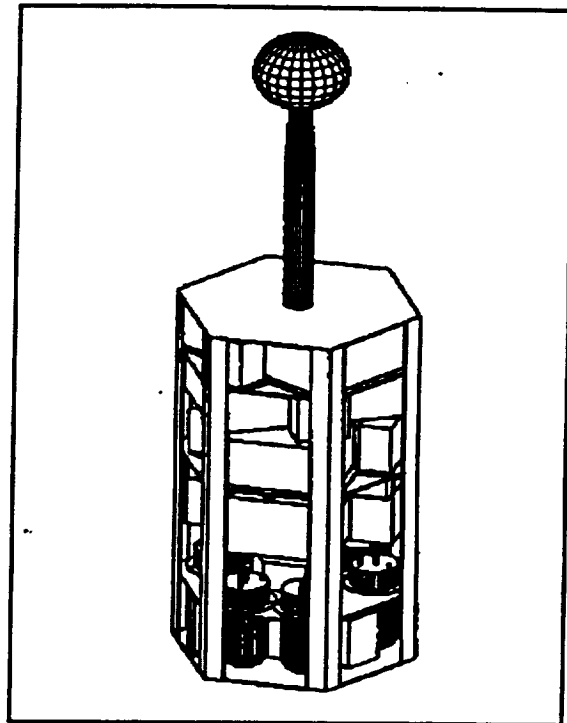


Figure 9. ISAT geometry

satellite was used to size the boom length and mass for this project. The current design configuration of ISAT is shown in Figure 9.

Current efforts of the Iowa Satellite Project provided much of the needed information about the satellite attitude dynamics. Mainly, the attitude determination and control group provided the mass moment inertias of the spacecraft body. These inertias assumed uniform mass distribution within the satellite. A design sizing code with boom length and tip mass as control variables was written to determine an optimum length and tip mass for the satellite. The code allows the user to select the desired stability in terms of the stability criterion Θ_x . The user also selects the desired range of boom lengths and tip masses. The code then iterates through these two variables, calculating the new mass moment of inertia I_x , and uses the new inertia to calculate the value of Θ_x . The calculated value of Θ_x is subtracted from the target value, and the absolute value of the difference is written to a data file along with the boom length and mass. This information can then be plotted on a contour plot.

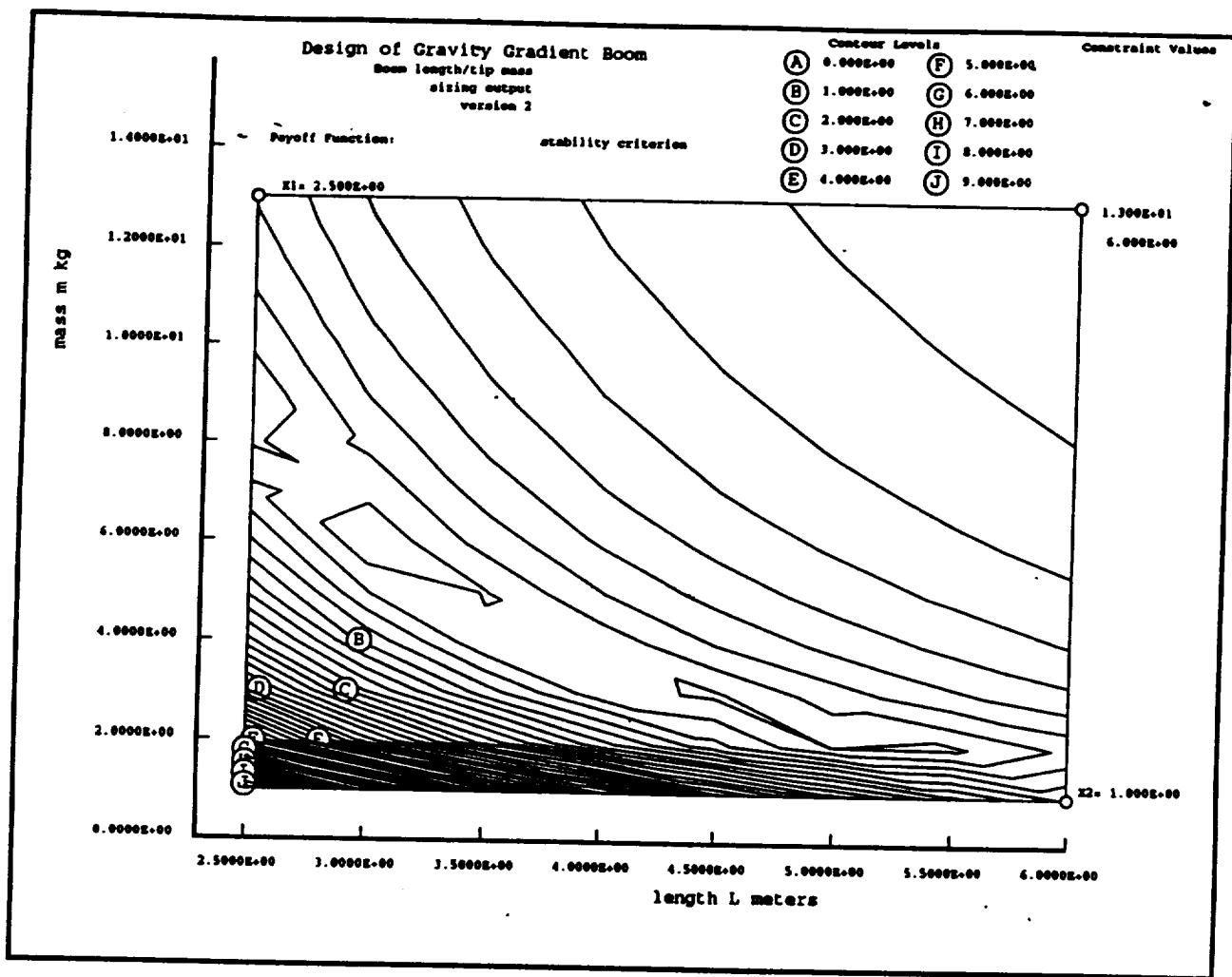


Figure 10. Boom Sizing Contour Plot

Figure 10 shows the contour plot for a stability criterion (Θ_s) of 0.98. Several candidate designs can be seen to meet the desired stability. The final boom size was selected to be 3 meters in length with a 6 kilogram tip mass. This represents almost a fifteen fold increase in I_p .

Stability considerations determined the boom length and tip mass. The diameter and thickness of each section would be determined according to the loads acting upon each section. Loads acting on the gravity-gradient boom were modeled as the Earth's gravitational force acting on the tip mass. A section sizing code was written to size the boom sections based on the known loads, the material properties of graphite-epoxy composites⁶, and a selected factor of safety. It was not possible to accurately determine

dynamic loads on the gravity-gradient boom due to insufficient source data from the ISAT project, so the factor of safety used in the sizing analysis was increased.

The code was set up to allow the user to choose the desired tip mass and boom length for section sizing analysis. The forces and moment on the boom were then determined based on the user's input of the tip mass and the maximum deflection of the satellite from the vertical axis. Once the user had completed the inputs of number of sections, minimum section thickness, gap between sections, and initial base section outer diameter, the code would size the boom sections. This entailed determining the necessary outer diameters, section thicknesses, and inner diameters corresponding to each boom section. If the

section thickness needed was found by the code to be smaller than the minimum input section thickness, the code would substitute in the minimum section thickness and continue the analysis. This measure was included in order to allow for more reasonable composite layup thicknesses in the fabrication process.

Results of section sizing yielded necessary section thicknesses for a number of base section outer diameters. A base section having an outer diameter of 3.81 centimeters was chosen based on the limited volume fraction of ISAT-1 which would be required. Table 1 shows the section sizes in terms of outer diameters, thicknesses, and corresponding inner diameters. It should be noted that in all cases the thicknesses used are the minimum section thickness, and the gap between sections is constant for ease of fabrication.

Boom Section (#)	Outer Dia. (cm)	Section Thickness (cm)	Inner Dia. (cm)
1	3.81	0.102	3.71
2	3.61	0.102	3.51
3	3.40	0.102	3.30
4	3.20	0.102	3.10
5	3.00	0.102	2.90
6	2.79	0.102	2.69

Table 1. Boom Section Sizes

Fabrication Methods

Fabrication methods were developed to produce the necessary sections to conform to the design. This included the construction of suitable mandrels and development of layup procedures. This also included developing a fabrication technique for the section joints.

The mandrels used for curing of the boom sections consisted of two sections of steel pipe, each 0.61 meters in length. The two pipes had inside diameters of 3.43 centimeters and 3.94 centimeters. These pipes were split along their length at the diameter. These were complemented by three rubber hose sections of 0.76 meters in length. The hoses had outside diameters 2.62 centimeters, 3.15 centimeters, and 3.61 meters. This provided the capability to

fabricate sections of varying diameter and thickness. The hoses were fitted internally with steel rods along their entire length. Both ends of the hoses were sealed, one containing a valve, to allow pressurization during the curing process.

Each section was fabricated from three plies of woven [0°/90°] graphite-epoxy composite. The composite material was cut to a length of 0.56 meters and a width equal to the circumference of the desired section. The three plies were then debulked with the edges staggered prior to being placed in the mandrel. The debulked composite was then placed in the mandrel and wrapped around the hose. Finally, the pipe section was clamped around it. Bleeder cloth was inserted between the composite material and the outer pipe as necessary to form each section.

With the mandrel clamped together, the inner hose was pressurized to 275,790.3 Pa (40 psi), and the mandrel was placed in an oven to cure for 3 1/2 hours at 176° C. Upon completion of the cure, the mandrel was disassembled, and the completed boom section was removed.

To avoid the added complexity and difficulties involved with co-curing the joints along with the respective boom sections, joints from separate cure cycles were epoxied in at a later time. A suitable gas seal was added to each section, and the boom was assembled.

Testing

Once the boom was complete, testing was conducted to ensure that the mechanical performance of the gravity-gradient boom design was satisfactory. The first test to be run was an analytical test involving ANSYS finite-element analysis. The boom was modeled on ANSYS using plate elements. A static analysis was performed. Figure 11 gives a representation of the deflection resulting in a maximum load condition applied to the boom and its tip mass. The boom was found to have a maximum deflection at the tip of 0.05 meters (1.969 inches).

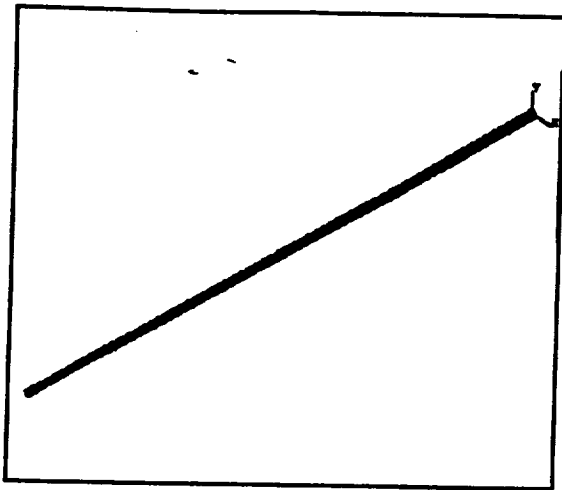


Figure 11. ANSYS Model

Recommendations

Further effort in this subject area is needed to enhance the understanding of the application of graphite-epoxy composites to gravity-gradient booms. The feasibility of the application has been demonstrated. However additional development work and testing is needed to provide the validation necessary to prepare hardware for use in a space application.

Additional testing of the boom section joint to verify the gas seal design and deployability is needed. The magnitude of gas charge needed for deployment in orbit is yet to be determined. Also, laboratory testing of the joint mechanics would serve to validate the length of the joint in respect to the boom sections.

Additional mechanical testing would serve to characterize the composite material properties obtained under the cure conditions used in this project. The repeatability of mechanical properties would build confidence in the production of flight hardware.

Thermal testing is needed to ascertain the optimum ply orientations to minimize thermal expansion coefficients. This testing was not undertaken previously due to inadequate facilities.

Summary

This project has demonstrated the feasibility for use of graphite-epoxy composites in gravity-gradient booms for small satellites. Given the mechanical and thermal properties of graphite-epoxy composites, a boom was designed and fabricated. The boom showed good attitude stability characteristics and encouraging mechanical behavior.

References

- ¹ Roberson, R. E., "Gravitational Torques on a Satellite Vehicle", *J. Franklin Inst.* 265, 12-33 (1958).
- ² *Spacecraft Attitude Dynamics and Control*, Vladimir A. Chobotov, Krieger Publishing Company, Malabar, Florida (1991).
- ³ *Modern Spacecraft Dynamics and Control*, Marshal H. Kaplan, John Wiley and Sons, Inc., New York, New York (1976).
- ⁴ Hunt, J.W., and Ray, J.C., "Flexible Booms, Momentum Wheels, and Subtle Gravity-Gradient Instabilities," AIAA Paper No. 92-1673 (1992).
- ⁵ Peterson, M.R., and Grant, D.G., "The Polar BEAR Spacecraft," *Johns Hopkins APL Tech. Dig.* Vol. 8, no. 3, 295-302 (1987).
- ⁶ *Composite Materials for Aircraft Structures*, Brian C. Hoskin and Alan A. Baker, Ed., American Institute of Aeronautics and Astronautics, Inc., New York, New York (1986).

Appendix C.3

Payload

Meteoroid and Debris Analysis for ISAT-1

Todd Kuper

May 21, 1993

Introduction

While in orbit, ISAT-1 will be in an environment filled with meteoroids and orbital debris. Within 2000 km of Earth's surface there is about 200 kg of meteoroid material, most of it only 0.1 mm in diameter.[1] In addition to this naturally occurring material there is estimated to be 1.5 to 3 million kg of man-made material as of mid-1988. Nearly 5000 objects with altitudes below 2000 km are continuously tracked by the U.S. Space Command Space Surveillance Center. Of these 5000 objects, only 6% are operational satellites and over 40% are fragments caused by breakups and collisions.[2] U.S. Space Command can only track objects that are 10 cm in diameter or larger; therefore, any estimates of debris population smaller than this comes from impact and fragmentation modeling and from studying satellites that have been retrieved from orbit such as the Long Duration Exposure Facility (LDEF). There is an estimated 1000 kg of debris with diameters less than 1 cm and about 300 kg of debris with diameters less than 1 mm. The greatest threat occurs at altitudes lower than 1000 km, or lower low Earth orbit (LEO1), the region where ISAT-1 will be operating. The population in LEO1 is growing by about 120 objects per year as of 1989. The population is expected to increase substantially in the next decade with the international space station, proliferation of micro-satellites, and new collisions and breakups. Due to an increase in the use of higher inclination orbits over the last decade, the average relative velocity of debris has also increased substantially and is expected to continue.[3]

Number of Impacts

The number of impacts on a spacecraft depends on the particle flux and the spacecraft's size, shape, and orientation. Flux is the number of objects impacting on a randomly tumbling surface per unit time and area averaged over time. Because the particle flux is usually calculated for a tumbling surface, the k-factor method is used to evaluate the expected number of impacts on a surface with a fixed orientation. The constant k is a function of the spacecraft's shape, as well as its zenith and azimuth angles. The zenith angle, θ , is the angle between the Earth vertical axis and the outward surface normal and the azimuth angle, ϕ , is the angle between the spacecraft's velocity vector and the outward surface normal. Values for the k-factor can be found from Figure 1.

The number of impacts on a spacecraft, N , during a period of time is given by the following expression

$$N = \int_{t_1}^{t_2} \sum_{i=1}^{n_s} k_i F_r A_i dt \quad (1)$$

where

- n_s = number of surfaces
- k = k-factor
- F_r = cumulative flux on a randomly tumbling surface
- A = surface area

Once N has been determined, the probability of exactly n impacts occurring in the corresponding time interval is found from Poisson statistics

$$P_n = \frac{N^n}{n!} e^{-N} \quad (2)$$

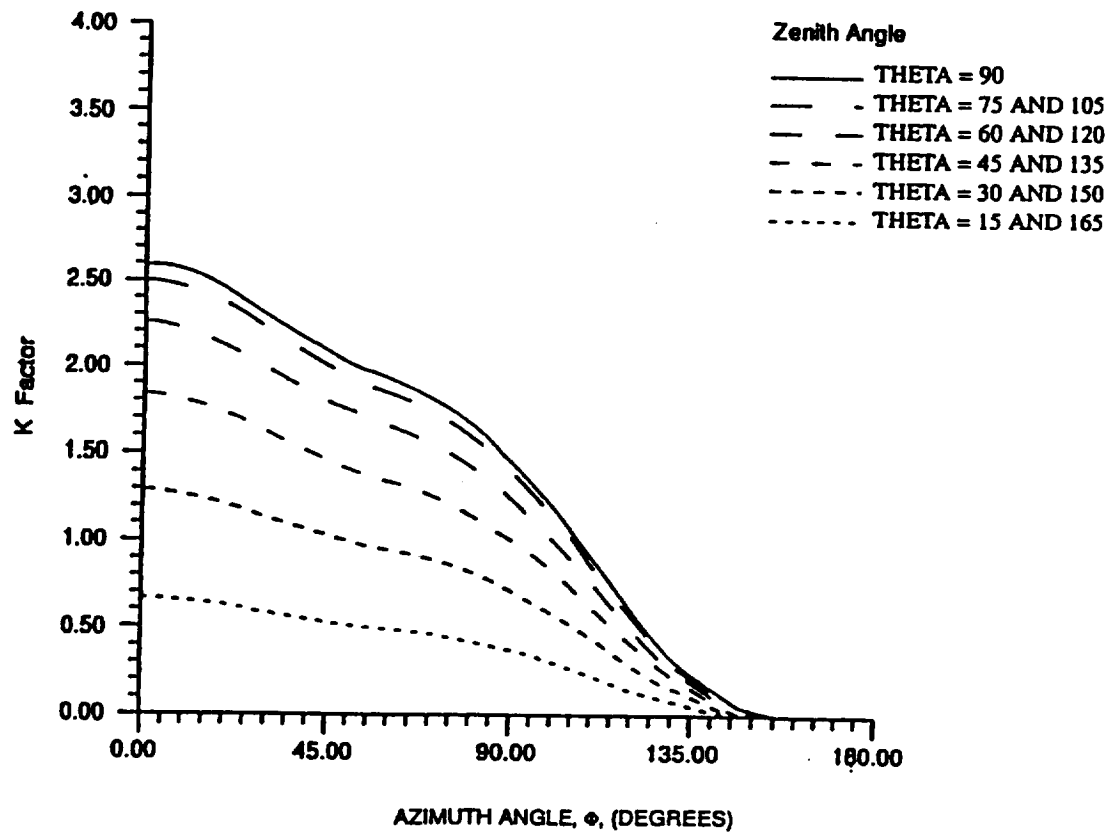


Figure 1: K-factor For Values of Zenith and Azimuth Angles

Meteoroids

Meteoroids that stay with their natural parent body and create periods of high flux are called streams. Random fluxes are called sporadic. The meteoroid environment that was used in this analysis was comprised of the average sporadic meteoroids and a yearly average of the stream meteoroids.

Flux

The cumulative meteoroid flux, F_r , for meteoroids of mass, m , is given by the following expression

$$F_r = s_f G_E F_r^{ip} \quad (3)$$

where F_r^{ip} is the interplanetary flux at 1 A.U. and is given by

$$F_r^{ip}(m) = c_0 \left[(c_1 m^{0.306} + c_2)^{-4.38} + c_3 (m + c_4 m^2 + c_5 m^4)^{-0.36} + c_6 (m + c_7 m^2)^{-0.85} \right] \quad (4)$$

where

$$\begin{aligned} c_0 &= 3.156 \times 10^7 \\ c_1 &= 2.2 \times 10^3 \\ c_2 &= 15 \\ c_3 &= 1.3 \times 10^{-9} \\ c_4 &= 10^{11} \\ c_5 &= 10^{27} \\ c_6 &= 1.3 \times 10^{-16} \\ c_7 &= 10^6 \end{aligned}$$

S_f is the Earth shielding factor and G_E is the focusing factor caused by the Earth's gravitational field. They can be calculated from the following expressions

$$S_f = \frac{(1 + \cos \eta)}{2} \quad (5)$$

$$G_E = 1 + \left(\frac{R_E}{r} \right) \quad (6)$$

and

$$\sin \eta = \frac{R_E}{(R_E + H)} \quad (7)$$

where

$$\begin{aligned} R_E &= \text{Earth's radius} + 100 \text{ km atmosphere (6478 km)} \\ H &= \text{height above atmosphere} \\ r &= \text{orbit radius} \end{aligned}$$

Mass Density

The estimated mass density for meteoroids, ρ , of mass, m , is given by the following expression

$$\rho \text{ (gm/cm}^3\text{)} = \begin{cases} 2 & m < 10^{-6} \text{ gm} \\ 1 & 10^{-6} < m < 0.01 \text{ gm} \\ 0.5 & m > 0.01 \text{ gm} \end{cases} \quad (8)$$

Velocity Distribution

Because of the precession of a satellite's orbit and the tilt of the Earth's equatorial plane with respect to the ecliptic plane, the meteoroid environment can be assumed to be omnidirectional relative to the Earth. The normalized velocity distribution with respect to the Earth is given by

$$f'(v) = \begin{cases} 0.112 & 11.1 < v < 16.3 \text{ km/s} \\ 3.328 \times 10^5 v^{-5.34} & 16.3 < v < 55 \text{ km/s} \\ 1.695 \times 10^{-4} & 55 < v < 72.2 \text{ km/s} \end{cases} \quad (9)$$

The normalized velocity distribution is the fraction of impacts in a 1 km/s interval and is given by the following expression

$$f'(v) = \frac{f(v)}{\int_0^\infty f(v) dv} \quad (10)$$

The velocity relative to the spacecraft is just the vector sum of the meteoroid's velocity relative to the Earth and the spacecraft's velocity relative to the Earth.

Orbital Debris

Flux

The cumulative flux, F_r , of debris of diameter d and larger on a randomly tumbling surface is given by the following equations for altitudes below 2000 km.

$$F_r = H\phi\Psi(F_1g_1 + F_2g_2) \quad (11)$$

where

$$H = \left[10^{\exp\left(-\frac{(\log d - 0.78)^2}{0.437^2}\right)} \right]^{\frac{1}{2}} \quad (12a)$$

$$\phi_1 = 10^{\left(\frac{1}{360} - \frac{1}{136} - 1.5\right)} \quad (12b)$$

$$\phi = \frac{\phi_1}{(\phi_1 + 1)} \quad (12c)$$

$$F_1 = 1.22 \times 10^{-5} d^{-2.5} \quad (12d)$$

$$F_2 = 8.1 \times 10^{10} (d + 700)^{-6} \quad (12e)$$

$$g_1 = \begin{cases} (1 + q)^{(t-1988)} & \text{for } t < 2011 \\ (1 + q)^{23} (1 + q')^{(t-2011)} & \text{for } t \geq 2011 \end{cases} \quad (12f)$$

$$g_2 = 1 + [p(t - 1988)] \quad (12g)$$

where

- d = debris diameter, cm
 t = year
 h = altitude, km
 S = solar radio flux, $F_{10.7}$, for t-1 year, 10^4 Jy
 p = assumed annual growth rate of intact objects in orbit, ($p = 0.05$)
 q and q' = estimated growth rate of fragments in orbit, ($q = 0.02, q' = 0.04$)

and Ψ is the inclination function given by Table 1.

Table 1: Inclination Function Values

Inclination, deg	Ψ
28.5	0.91
30.0	0.92
40.0	0.96
50.0	1.02
60.0	1.09
70.0	1.26
80.0	1.71
90.0	1.37
100.0	1.78
120.0	1.18

Mass Density

Knowledge of debris shape and density is very scant. For the purposes of this analysis, the objects were assumed to be spherical. The average density, ρ , for an object with diameter, d , can be taken as

$$\rho \text{ (gm/cm}^3\text{)} = \begin{cases} 4 & \text{for } d < 0.62 \text{ cm} \\ 2.8d^{-0.74} & \text{for } d \geq 0.62 \text{ cm} \end{cases} \quad (13)$$

Velocity Distribution

The collision velocity distribution is the number of impacts with velocities between v and $v + dv$ relative to the spacecraft. Averaged over all altitudes, the distribution is given by

$$f(v) = \{2vv_o - v^2\} \left\{ G \exp \left[- \left(\frac{v - Av_o}{Bv_o} \right)^2 \right] + F \exp \left[- \left(\frac{v - Dv_o}{Ev_o} \right)^2 \right] \right\} + HC (4vv_o - v^2) \quad (14)$$

where

$$\begin{aligned}
A &= 2.5 \\
B &= \begin{cases} 0.5 & i < 60 \\ 0.5 - 0.01(i - 60) & 60 < i < 80 \\ 0.3 & i > 80 \end{cases} \\
C &= \begin{cases} 0.0125 & i < 100 \\ 0.0125 + 0.00125(i - 100) & i > 100 \end{cases} \\
D &= 1.3 - 0.01(i - 30) \\
E &= 0.55 + 0.005(i - 30) \\
F &= \begin{cases} 0.3 + 0.0008(i - 50)^2 & i < 50 \\ 0.3 - 0.01(i - 50) & 50 < i < 80 \\ 0.0 & i > 80 \end{cases} \\
G &= \begin{cases} 18.7 & i < 60 \\ 18.7 + 0.0289(i - 90)^3 & 60 < i < 80 \\ 250.0 & i > 80 \end{cases} \\
H &= 1.0 - 0.0000757(i - 60)^2 \\
v_o &= \begin{cases} 7.25 + 0.015(i - 30) & i < 60 \\ 7.7 & i > 60 \end{cases} \quad (15)
\end{aligned}$$

and i is the orbit inclination in degrees. When $f(v)$ is negative, the function is to be set to zero. The normalized velocity distribution is the fraction of impacts over a 1 km/s interval and is given by the following expression

$$f'(v) = \frac{f(v)}{\int_0^\infty f(v) dv} \quad (16)$$

Results

It was assumed for this study that ISAT-1 will be orbiting with a perfect attitude, i.e. non-rotating and the gravity-gradient boom will be pointing exactly towards zenith. With this assumption, the zenith angle will be 90 degrees. Figure 2 shows the azimuth angle for the six lateral surfaces on ISAT-1. Using Figure 1, the k-factor for ISAT-1 was found to be 7.80. The surface area was $0.10931m^2$ for each of ISAT-1's six lateral sides. It was also assumed that ISAT-1 will be in an orbit with an altitude between 500 and 1000 km and an inclination of 60 degrees.

Figure 3 shows the expected number of impacts on ISAT-1 from meteoroids of varying mass during five years in orbit. It can be seen that only one impact from meteoroids of 0.1 mg and larger is expected. At the same time, meteoroids of 1 μg and larger are expected to impact ISAT-1 about 5.5 times per day. Figure 4 shows that most meteoroids have velocities between 12 and 16 km/s relative to the Earth. A spacecraft in a 600 km circular orbit will be traveling at 7.6 km/s. Inclination and altitude have little effect on the meteoroid flux or velocity distribution.

Figures 5 and 6 show the number of impacts from orbital debris and the debris's velocity distribution, respectively. It can be seen from Figure 5 that only one impact is expected from debris with diameters 0.3 mm and larger. Using equation 13 the mass for a 0.3 mm piece of debris will be about 0.45 mg. It is also expected that debris with diameters of 10 μm and larger (17 ng and larger) will impact ISAT-1 about once per day. Figure 6 shows that most orbital debris will impact ISAT-1 with velocities between 12 and 15 km/s for an inclination of 60 degrees.

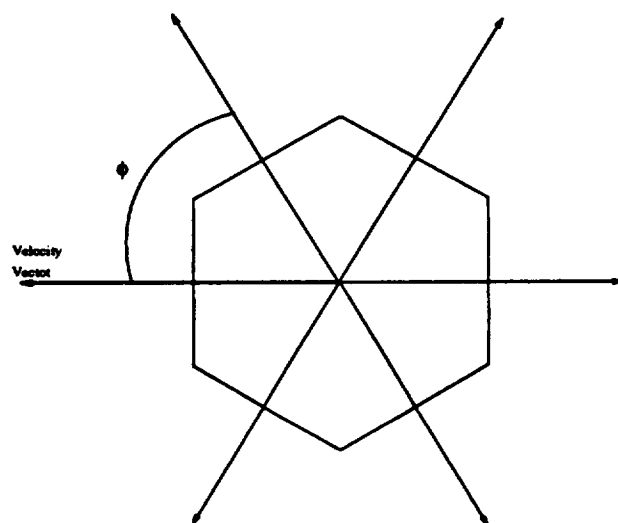


Figure 2: Azimuth Angle Geometry of ISAT-1

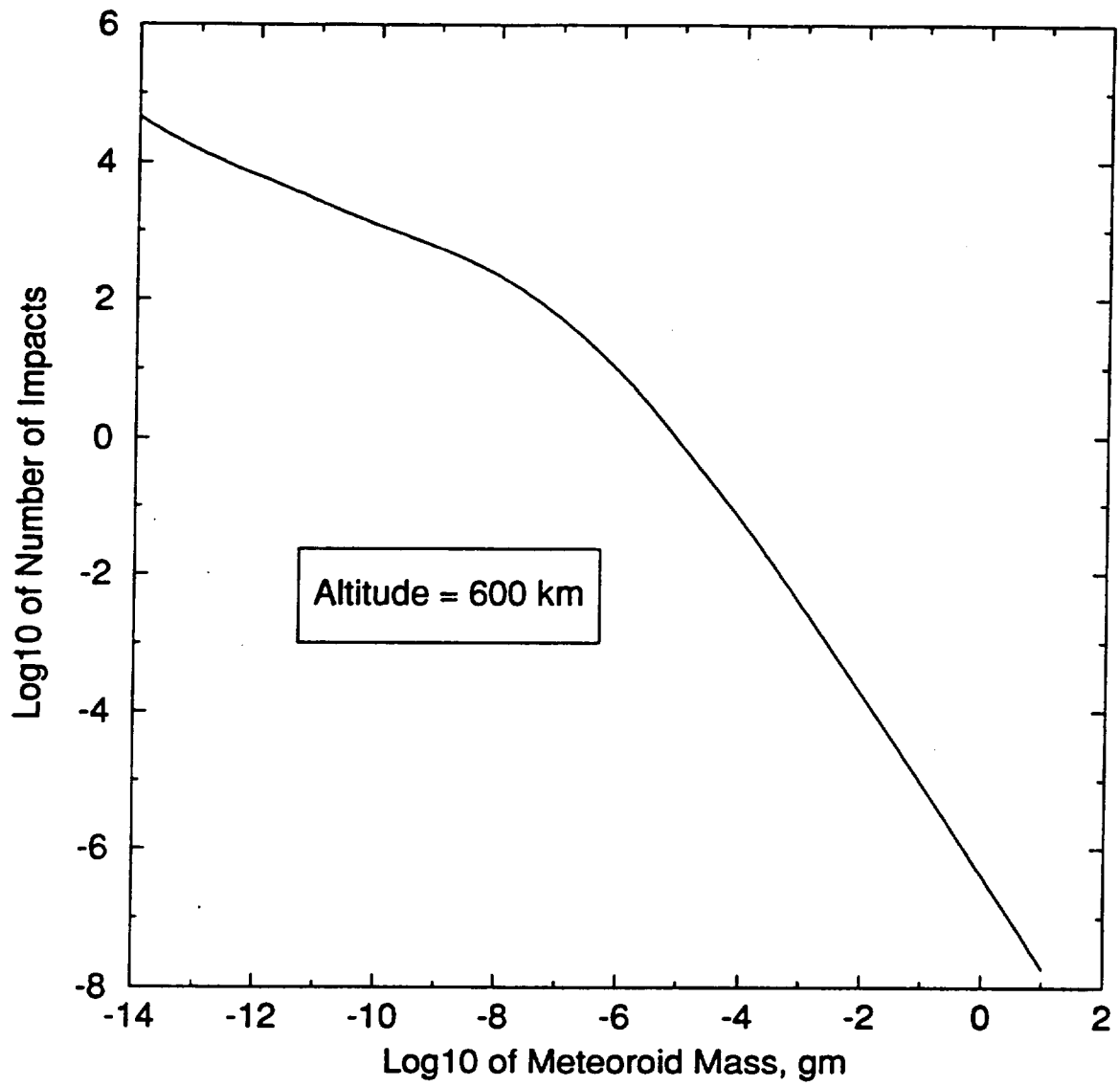


Figure 3: Expected Number of Impacts on ISAT-1 From Meteoroids During 5 Years

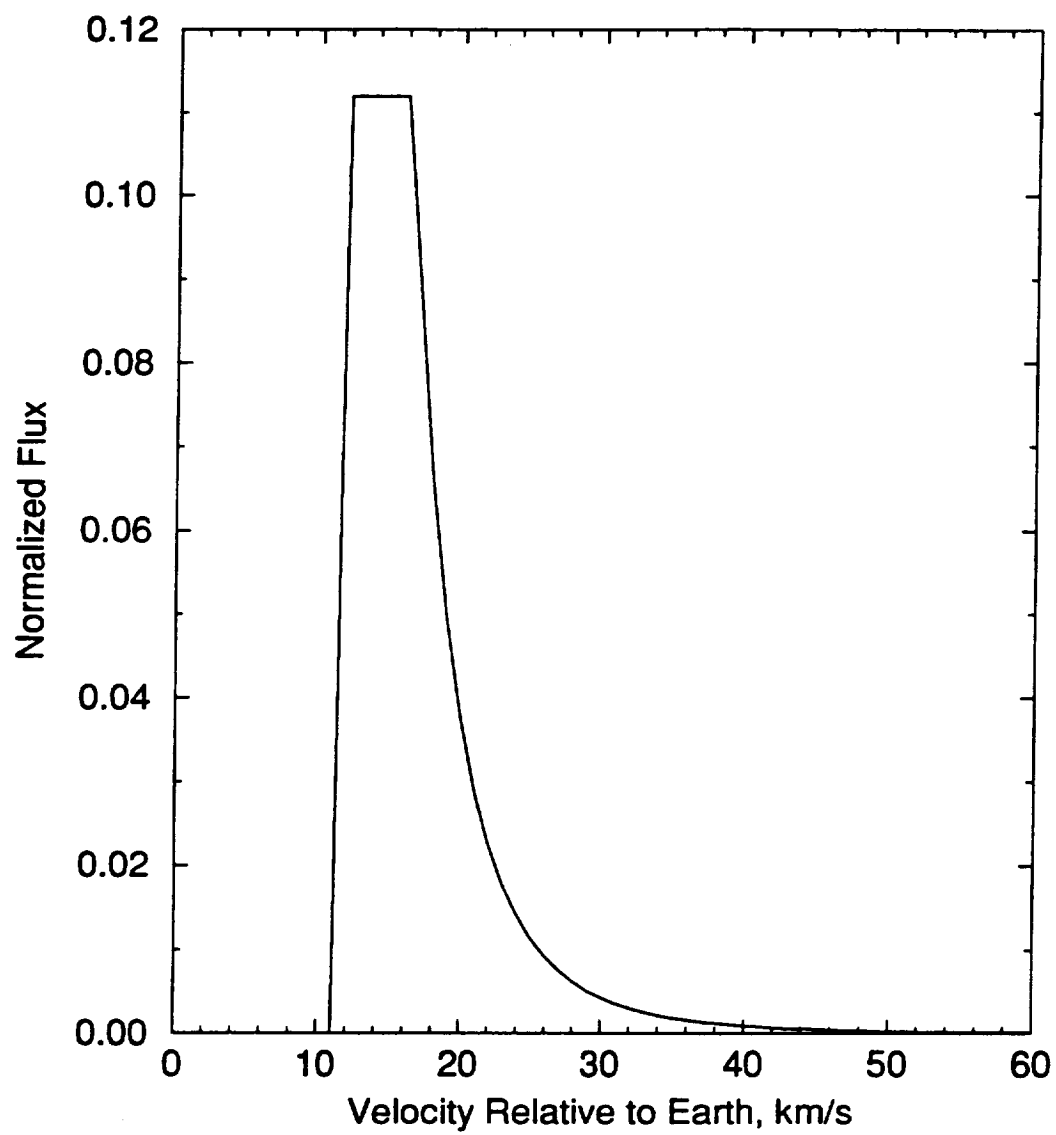


Figure 4: Normalized Meteoroid Velocity Distribution

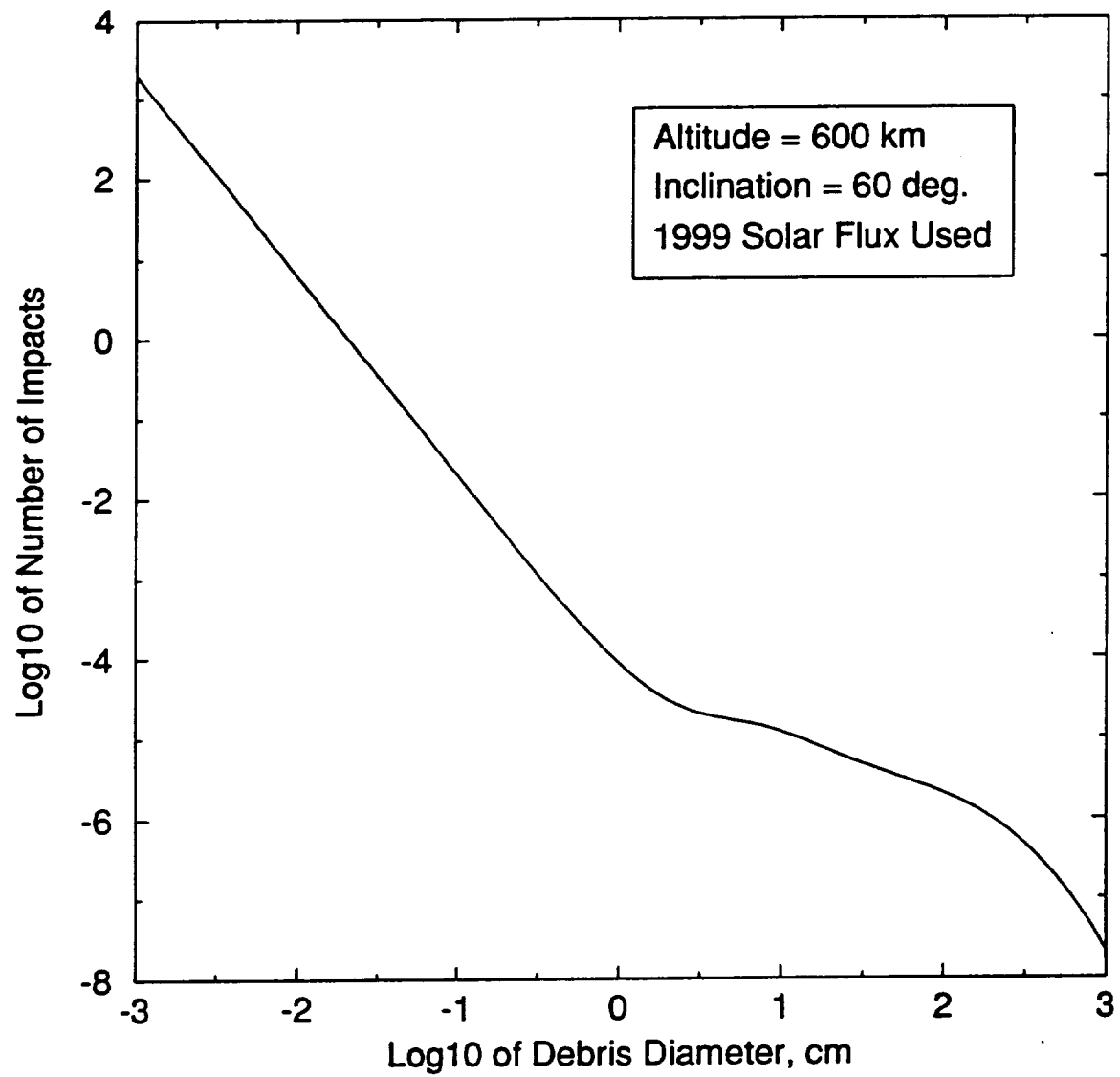


Figure 5: Expected Number of Impacts on ISAT-1 From Orbital Debris During 5 Years

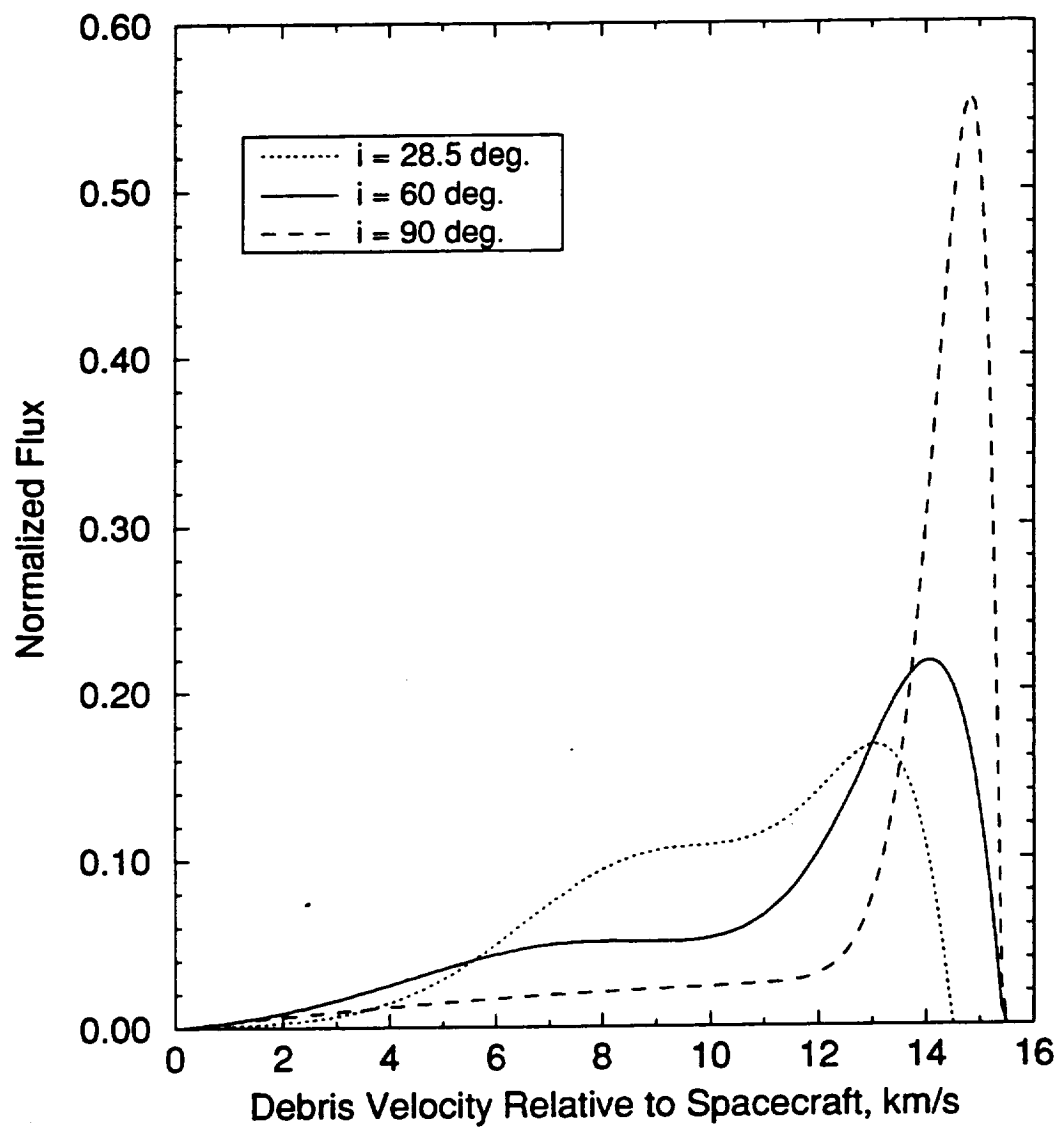


Figure 6: Normalized Orbit Debris Velocity Distribution

References

- [1] SSP 30425 Revision A, June 1991, "Space Station Program Natural Environment Definition For Design."
- [2] McKnight, D.S., and Johnson, N.L., "Catalog Growth Rate Study (Hazard Analyzed in Geosynchronous Transfer Orbits)," *Journal of Spacecraft and Rockets*, Vol. 29, 1992, pp.64-69.
- [3] McKnight, D.S., and Anz-Meador, P.D., "Historical Growth of Quantities Affecting On-Orbit Collision Hazard," *Journal of Spacecraft and Rockets*, Vol. 30, 1993, pp. 120-124.

Appendix C.4

Power Systems

Solar Array Power Generation Analysis for ISAT-1

Todd Kuper

June 1, 1993

Introduction

ISAT-1 will obtain the necessary power to run the bus and payload subsystems by solar arrays body-mounted on the lateral faces of the bus. This analysis was done to find out how much power can be generated as a function of orbit altitude, sun elevation above the orbit plane, ISAT-1's position in its orbit about the Earth, and the efficiency of the solar array.

Coordinate System

The analysis was done using Earth-referenced spacecraft-centered celestial coordinates as shown in Figures 1 and 2. The orbit plane is the equator of the globe plot and the direction of the Earth, the yaw direction, is held fixed. The Earth's disk is the shaded circle in Figure 2. Because the yaw axis is always facing nadir, this coordinate frame rotates once per orbit in inertial space about the orbit pole; therefore, any objects that are essentially fixed in inertial space during one orbit, such as the sun or moon, will appear to rotate about the orbit pole once per orbit. An eclipse will occur when the sun passes behind the disk of the Earth. As the Earth travels in its orbit about the sun and the longitude of the ascending node of the spacecraft's orbit rotates about the Earth's pole due to perturbations, the sun will travel up and down on the globe plot from a maximum of the inclination of the spacecraft's orbit plus the obliquity of the ecliptic above the orbit plane to this maximum below the orbit plane and back up to this maximum above the orbit plane during one year. For example, if the spacecraft's orbit has an inclination of 60 degrees, the sun will travel from a maximum elevation of 83.4 degrees above the orbit plane to 83.4 degrees below the orbit plane and then back up to 83.4 degrees above the orbit plane over the course of one year.

In this coordinate system the normal to a given face is represented by point F on the celestial sphere shown in Figure 3. The elevation of the sun above the orbit plane is represented by β_S and the elevation of point F above the orbit plane is represented by β_F . Az_S is the azimuth of the sun with respect to nadir and Az_F is the azimuth of point F with respect to nadir. The rotation angle ΔAz is the difference in azimuths between the sun and point F and varies uniformly from 0 to 360 degrees for a circular orbit. The angle between point F and the sun is represented by β . The incident solar energy on face F is then

$$I = AK e_f \cos \beta \quad (1)$$

where

- A = area of face F
- K = solar radiation constant (1358 W/m^2)
- e_f = overall efficiency of solar panel

and from the law of cosines for spherical trigonometry

$$\cos \beta = \cos \beta_F \cos \beta_S + \sin \beta_F \sin \beta_S \cos (\Delta Az) \quad (2)$$

When $\beta > 90$ degrees the sun is on the "back side" of face F and will not cause any power generation. Eclipses occur at sun azimuths given by

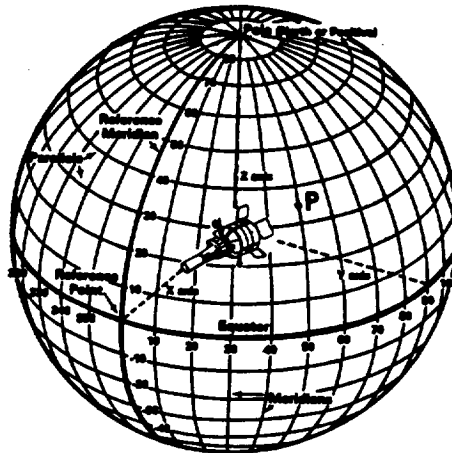
$$Az_{eclipse} = \arccos \left(\frac{\cos \rho}{\cos \beta_S} \right) \quad (3)$$

where

$$\sin \rho = \frac{R_E}{R_E + H} \quad (4)$$

where

ρ = angular radius of a spherical Earth
 R_E = radius of the Earth (6378.14 km)
 H = spacecraft altitude



Definition of a Spherical Coordinate System on the Unit Sphere. The point P is at an azimuth of 50 deg and elevation of 35 deg, normally written as $(50^\circ, 35^\circ)$.

Figure 1:

Results

ISAT-1 will be a hexagonal cylinder with the solar panels mounted on the six lateral sides. Five sides will have 0.1024 m^2 of solar cells while the side with the launch vehicle adapter will have 0.512 m^2 of solar cells. Because not all six sides have the same solar cell area an average power was calculated at each point in the orbit. This was done by "rotating" the model about the longitudinal axis through 360 degrees by steps of 5 degrees and averaging the power generated. This technique was used for generating Figures 4, 5, and 6.

Figure 4 shows the average amount of power generated in each orbit as a function of sun elevation and solar array efficiency at an altitude of 600 km. The average power at each sun elevation was determined by calculating the power generated at 30 second intervals during one orbit, including eclipse times, and then taking the average of the power generated. Altitudes between 400 km and 1500 km affect the average power generated very minimally. The linear decrease in power for lower efficiencies is from the linear relation in Equation 1.

Figure 5 shows the power generation profile during one orbit for an efficiency of 0.18 and at an altitude of 600 km. As can be seen from this figure, sun elevations near 0 degrees cause great fluctuations in the power generated at different points in the orbit. This comes from the fact that

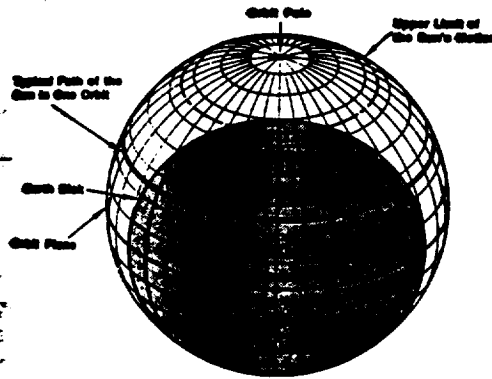


Figure 2:

the sun is shining mainly on the nadir and zenith ends of the spacecraft at azimuths near 0 and 180 degrees. When the elevation of the sun nears 90 degrees (or -90 degrees) the ends of the spacecraft are never in sunlight at any time during the orbit and the sunlight is concentrated on three of the lateral sides. This causes the power generation curve to become more flat. Also at higher sun elevations, the eclipse times become shorter. This is also reflected in Figure 7.

Figure 6 shows the average amount of energy generated during one orbit. This was calculated by numerically integrating the power generated during one orbit with time. As can be seen there is more energy generated at higher altitudes. This occurs because the eclipse times are shorter at higher altitudes. The curves in Figure 6 closely match the curves in Figure 4.

Figure 7 shows the time a satellite spends in sunlight and in eclipse as a function of altitude and sun elevation.

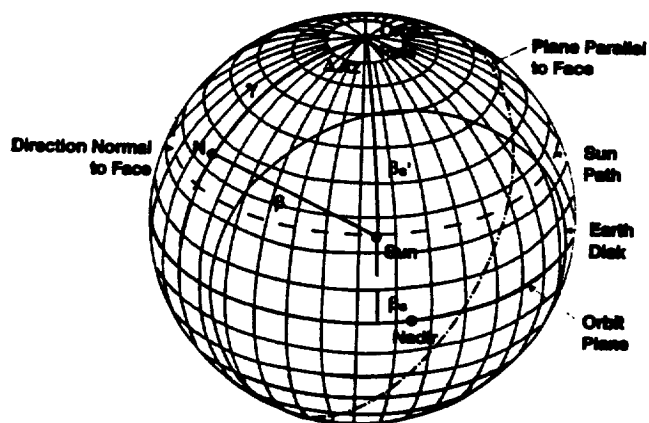


Figure 3:

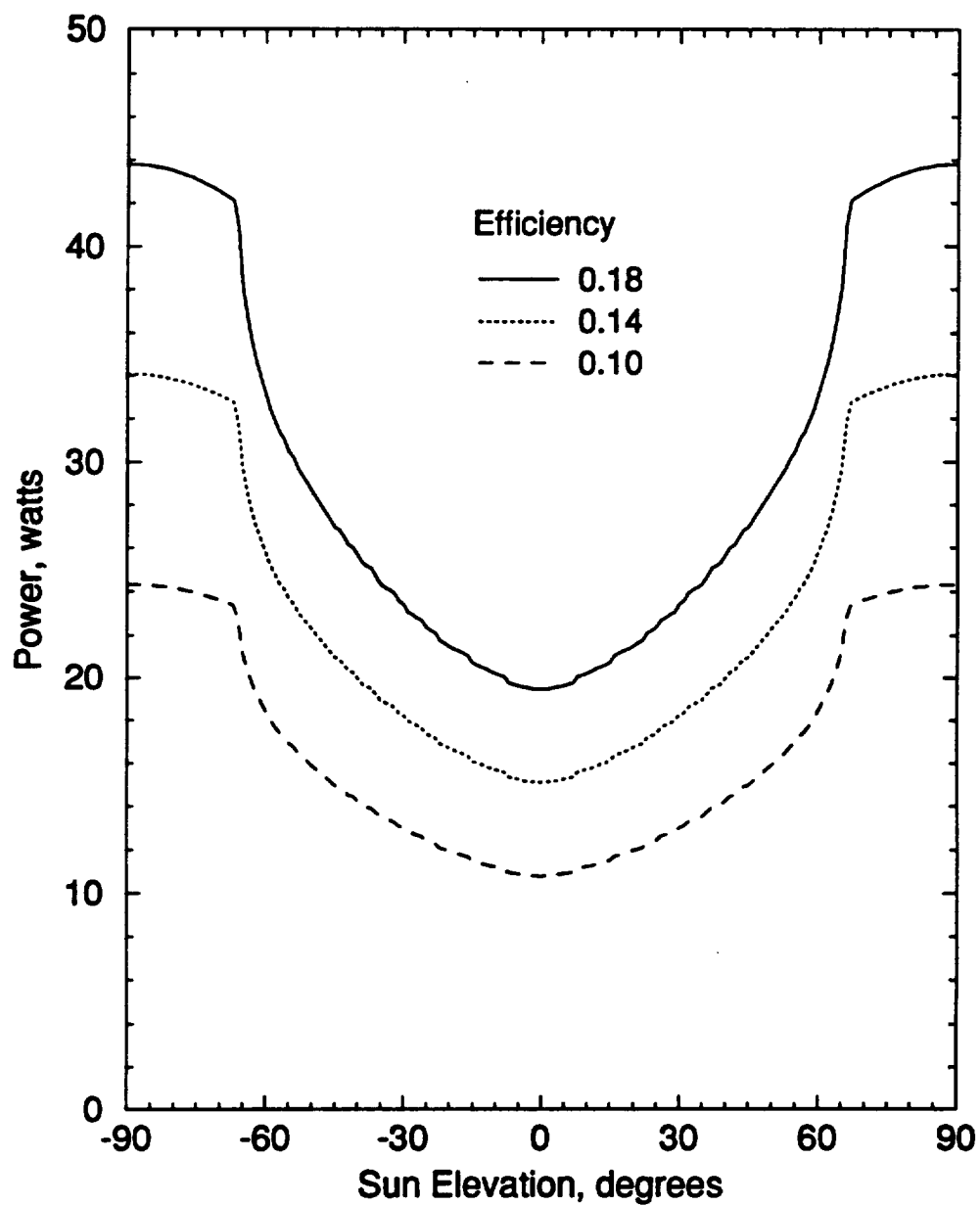


Figure 4: Average Power Generated During One Orbit at an Altitude of 600 km

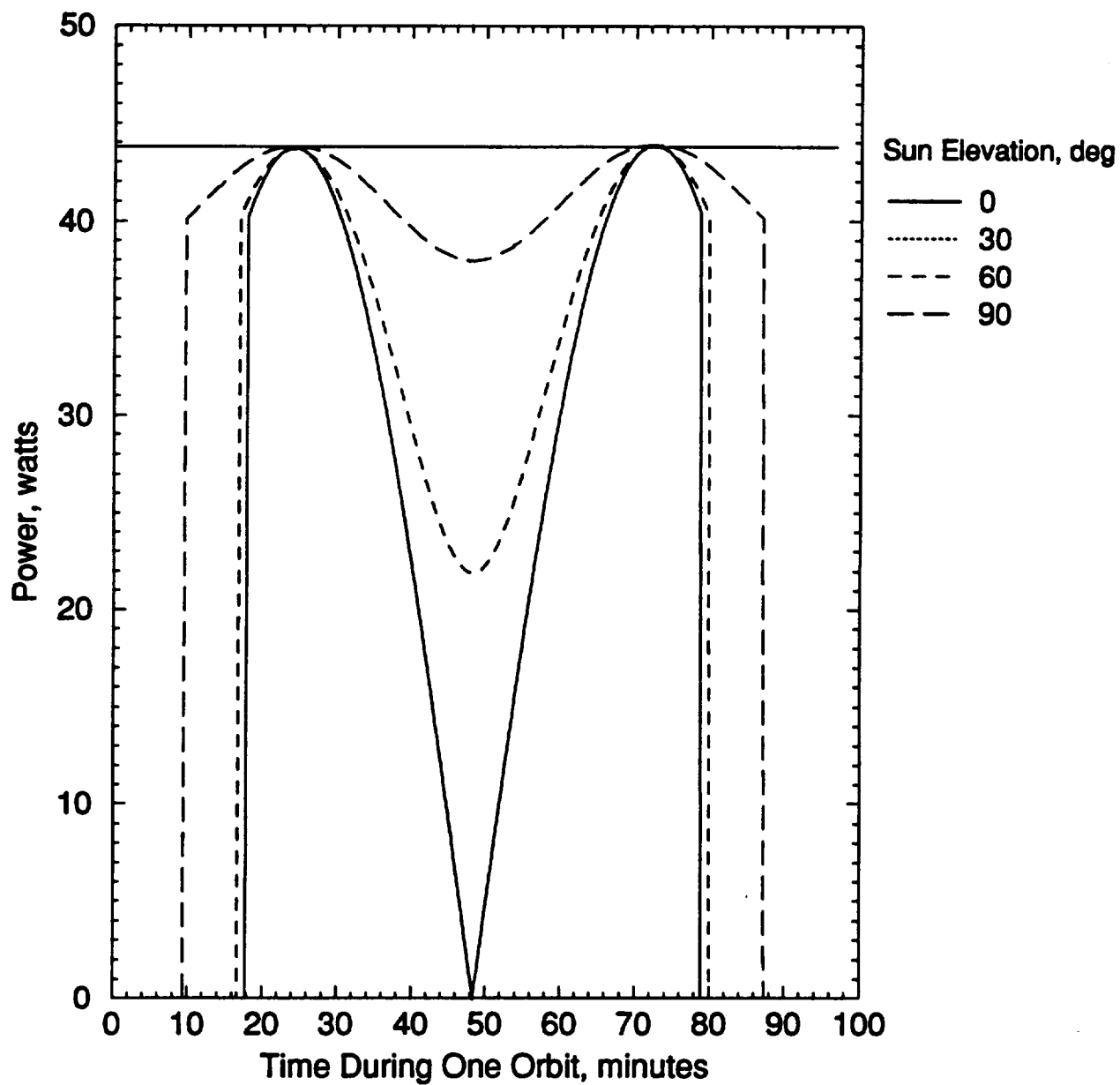


Figure 5: Power Generation Profile For One Orbit at an Efficiency of 0.18 and Altitude of 600 km

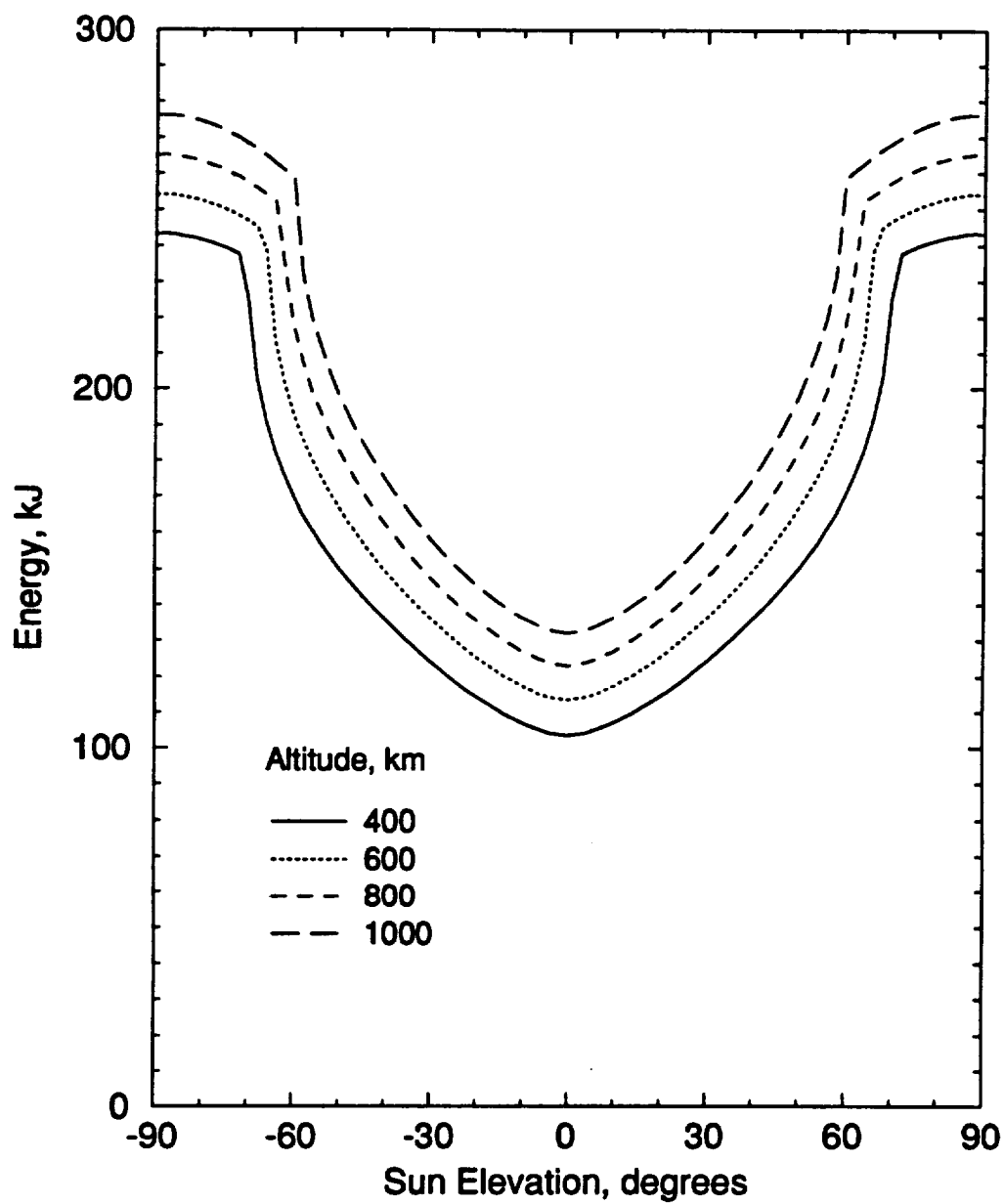


Figure 6: Average Energy Generated During One Orbit at an Efficiency of 0.18

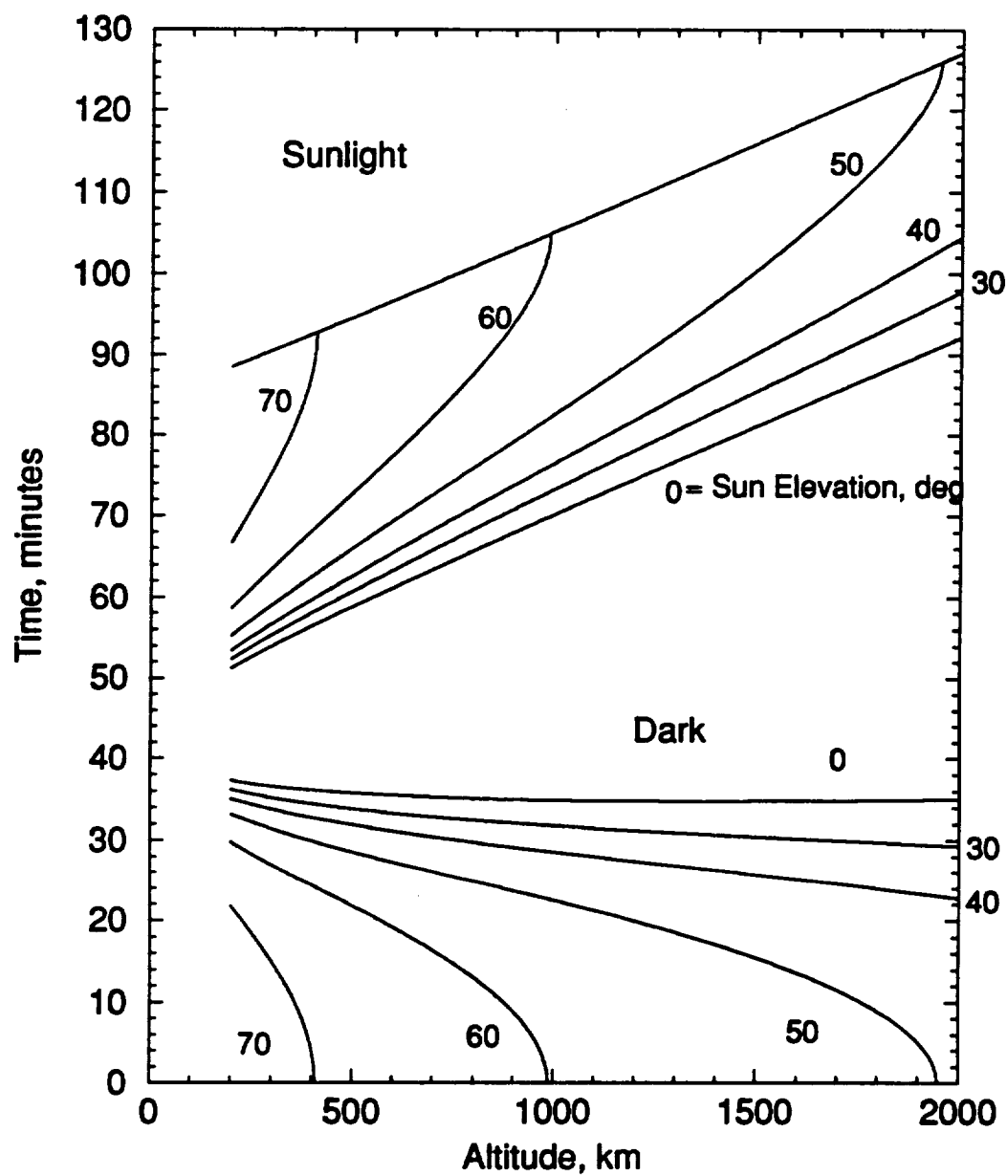


Figure 7: Time Spent in Eclipse and Sunlight

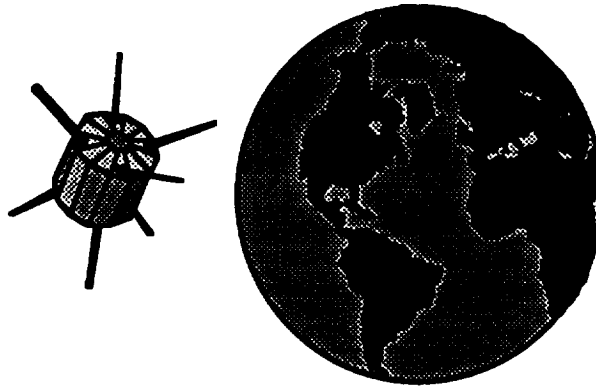
Appendix C.5

Thermal Considerations

58:056 Mechanical Engineering Project

Final Report:

**Design of a Thermal Control System for the ISAT-1
Satellite**



Presented By:

Group G

**Joseph Clay
David Prall
Matthew A. Schneider
Mike Skrbich**

**Mechanical Engineering Students
Department of Mechanical Engineering
The University of Iowa
Iowa City, Ia 52242**

Presented to:

**Prof. R.L. Stephens, PhD, P.E.
Department of Mechanical Engineering
The University of Iowa
Iowa City, Ia 52242**

May 10, 1993

ABSTRACT

ISAT-1 is a satellite being designed, built, and tested by students from the state universities of Iowa. The objective of this project was to design a method for determining the thermal control needs of the ISAT-1 Satellite. Space is a harsh environment with a temperature of four degrees kelvin, and the only method for heat transfer in space is through radiation. A satellite in space will be subjected to four main types of radiation: direct solar radiation, albedo radiation or solar radiation reflected from the earth, earth infrared radiation, and emission from the satellite itself. As the satellite orbits the earth, these different radiations will fluctuate causing fluctuations in temperature of the satellite. The fact that not all equipment on a satellite can operate at any temperature necessitates a method for controlling the temperatures seen by the satellite. This method is called thermal control.

In order to solve the problem of determining the thermal control needs of the ISAT-1 Satellite, a computer program was generated. This computer program was designed to be used for any satellite configuration by having the user input the properties of the satellite to be analyzed. The program works by splitting up the satellite into nodes which are representative of different parts of the satellite and pieces of equipment on the satellite. By inputting orbit information of the satellite, space environment information, and the thermal properties of the nodes of the satellite, the program simulates the temperatures the satellite will see during its orbit. These temperatures are output in the form of high, low, and mean temperature for each node during the orbit. Output data on the transient response for each node can also be obtained from the program to be used for graphing. The thermal control system in this program is represented by thermal resistances between each node which are input to the program. By inputting different values for the thermal resistances (R-values), the temperature of the nodes on the satellite can be controlled. Once the R-values that keep each node within its respective temperature tolerance are found, they can be used to choose materials, such as insulators or conductors, that will be used for the thermal control system of the satellite.

Not enough specific information about the ISAT-1 Satellite was known to perform and in depth analysis using this program. However, with the general configuration of the satellite, properties were assumed, and the program was ran to determine if it functioned properly. The results showed that the program worked, and it is suggested, that as more information about the ISAT-1 Satellite is obtained, the program is used to find the thermal control needs of the satellite.

ACKNOWLEDGMENTS

**We would like to thank the following people for graciously assisting us
in obtaining information essential to this project:**

**Prof. P.B. Butler
Department of Mechanical Engineering
The University of Iowa
Iowa City, Ia 52242**

**Bill Fisher
AeroSpace Company, The
Los Angeles, Ca 90216**

**Dr. Leverne K. Seversike
Iowa State Satellite Project Coordinator
Iowa State University
Ames, Iowa**

**Prof. T.F. Smith
Department of Mechanical Engineering
The University of Iowa
Iowa City, Ia 52242**

TABLE OF CONTENTS

	Page
RECOMENDATIONS.....	1
CONCLUSIONS.....	2
INTRODUCTION.....	3
THEORY.....	5
COMPUTER PROGRAM.....	12
RESULTS AND DISCUSSION.....	17
REFERENCES.....	21
APPENDICES.....	22

Appendix A: Background Information
Appendix B: Computer Program Handbook

RECOMMENDATIONS

Engineers use mathematics to understand and predict behavior in the physical world. If a mathematical model created fails to do so, then engineers are no longer engineers -- they are simply mathematicians. Therefore, it is desirable to verify a model in order to insure both its relevance to and accurate description of the system for which it was created. An improper model will fail both to fully describe the chosen system and to predict its behavior; thus, the general rationale for model verification is obvious.

The particular rationale for verification of the Thermal Math Model (TMM) for the ISAT-1 can be summarized as follows. It is desired to demonstrate that the principal heat transfer paths and pertinent heat transfer mechanisms have been correctly anticipated and understood. Model verification is also necessary in order to check the validity of the assumptions made concerning surface properties, thermal interfaces, and insulation effectiveness. Furthermore, it is important to verify the accuracy of the predictions that the model makes as these will be used to determine performance reliability.

Three principal methods exist for TMM verification: verification by independent analysis, verification with flight data, and verification by test. Verification by independent analysis involves a separate design team conducting a parallel analysis of the system. In this particular case, this method is not an option because this team is the only team which will be involved with this phase of the design for ISAT-1. In large scale satellite operations, often several, subsequent vehicles will be launched which are similar to one another in design and/or orbit. This allows for the establishment of a performance history constructed from the knowledge gained through flight data. ISAT-1 is the first of its particular design to be launched; therefore, there is no previously accumulated flight data. Due to this fact, verification with flight data is not a possibility. The remaining method is verification by test. This has become the chosen method for verifying the TMM of ISAT-1.

Two chief test types have been selected by this team to be recommended to the ISAT-1 committee. They are namely, a Thermal Balance Test and a Thermal Vacuum Test.

Detailed testing procedures have as yet to be decided. However, the following basic information concerning the tests has been acquired.

In effect, the thermal balance test is the qualification of the Thermal Control System (TCS) as it verifies the analytical thermal model. This test will be used to demonstrate the functional capability of the on-board thermal control hardware. Thermal balance testing is vital in establishing the ability of the TCS to maintain components, subsystems, and space vehicles within specified temperature tolerances for critical mission phases.

A thermal vacuum test is not the same as a thermal balance test. While the balance test is designed to simulate actual mission operation, the vacuum test is merely a hot / cold - soaking cycle which is used to subject the satellite to its anticipated environmental extremes in a worst-case scenario.

At present very little information is available concerning the payload for the satellite. In order for the design, or the computer program to be fully implemented, more specific data must be defined concerning ISAT-1. Complete information needs to be compiled which must include detailed information for all the intended on-board components (size, location, material composition, mass, power requirements, thermal tolerances, and duty cycles).

Once this information is known, the existing nodal network can be expanded to accommodate the true configuration for ISAT-1. The design described herein can then be used to generate the solution for the Thermal Control System of the satellite. The solution will contain all the TCS requirements, specifically the R-values, which can be manipulated through material / device selection to satisfy the thermal criteria for the payload components.

CONCLUSIONS

To date the design team for the TCS of ISAT-1 has succeeded in producing a design to determine satellite thermal control needs. Any configuration can be

accommodated, and the design has the capability to determine the TCS requirements for the particular configuration chosen. The requirements are in the form of R-values for thermal resistance.

In this preliminary analysis, a hypothetical configuration based upon the limited component information currently available has been used. The primary intent of this analysis was to prove that the present design is capable of producing accurate and meaningful results. The R-values are indeed meaningful as they represent design elements which can be effectively generated and modified to control and regulate thermal criteria for the components. The accuracy of the R-values lies in the fact that they have been derived from the appropriate mathematical models in which all the relevant physical phenomena were considered.

A major accomplishment of the design and analysis is that the need for a TCS for ISAT-1 has been confirmed. The analysis run for the hypothetical configuration has indicated the existence of potential problem areas with regard to thermal overload. The computer program has shown its ability to provide a solution to this problem by generating the R-values necessary to control the thermal overloads through appropriate manipulation (i.e., material / device selection and component location).

INTRODUCTION

The Iowa Satellite Project, ISAT-1 will not just be another communications satellite, it will be an educational tool. The ISAT-1 will be designed, built, operated, and used by students, and the general public throughout Iowa. Although all three of the Iowa Universities will be involved in the project, only the University of Iowa and Iowa State University will design the satellite. The University of Northern Iowa's main function will be to provide educational material and projects from the satellite information for use by

grade school age students and up. Iowa State will design orbit and the space systems part of the project, while University of Iowa will be designing the thermal control system or TCS for the satellite. More background about the satellite, including the equipment and experiments on board the satellite can be found in the Appendix A.

The objective of this project was to analyze the temperatures the ISAT-1 satellite will be subjected to, and then design the method for determining the thermal control needs of the satellite.

The thermal control system of a satellite is needed in order to maintain structural and equipment integrity over long periods of time. The fundamental requirement of the thermal control system is to allow optimum performance of components by controlling the temperature in which they operate. If it were feasible to operate the equipment and experiments at any temperature, there would be no need for a thermal control system. Reliable long term performance of most spacecraft components takes place at or near room temperature (293 K). Most of the equipment on board the satellite have certain temperature ranges that they can operate within, without thermal failure.(Fisher, 1993) However, as the satellite orbits the earth, the temperature in space is 4 K. This does not mean the satellite will be at 4 K because it will be subjected to different heat fluxes.

The only means of heat transfer in space is through radiation and conduction. There are four types of radiation that a satellite will be subjected to in space: radiation from the sun, albedo radiation (reflected sunlight off the earth), earth shine (infrared radiation from earth), and satellite emission. It can be seen from Figure 1, that the radiation from the sun and the albedo radiation will vary as the satellite passes in and out of the shadow of the earth. Therefore, the satellite's temperature will fluctuate with time. These fluctuations in temperature as well as the temperature in space make up the need for a thermal control system of a satellite.

HEAT TRANSFER BY RADIATION

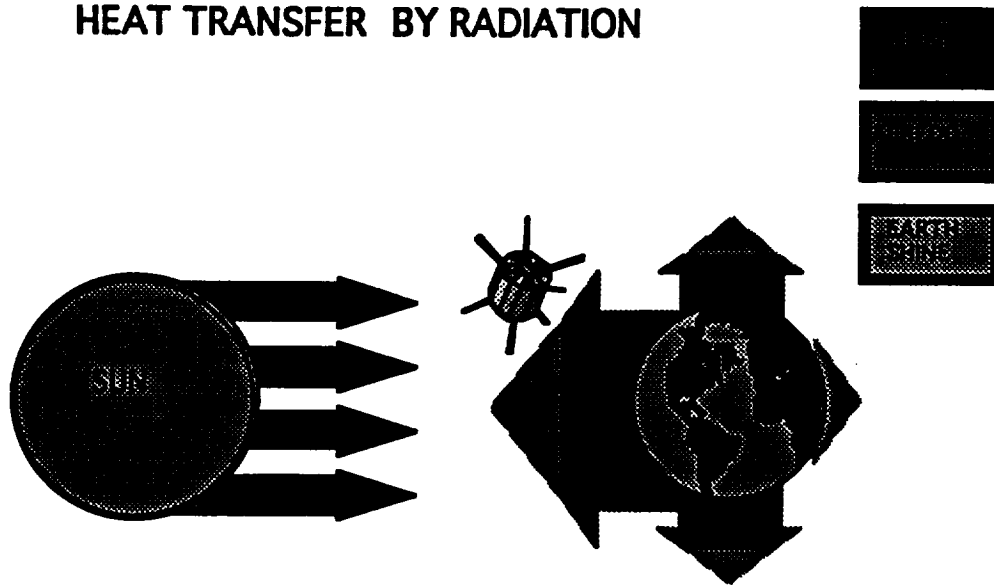


Figure 1: Radiation in the Space Environment

Design of the thermal control system for the ISAT-1 can be divided into five parts: define the spacecraft environment, define thermal control options, select and analyze thermal configuration, and select components. (Iowa, 1992) In order to perform a full analysis, concurrent engineering has had to take place between the state universities, and, as information is lacking due to this aspect of the project, personal goals for the purpose of the class have been set. Some information on the satellite's structure and components has been obtained, and it was decided to use this information to begin an analysis of the temperatures the satellite would see in space. To perform this, a computer program has been written to simulate the satellite in orbit. Based on the satellite's material properties the program analyses the temperatures of the components of the satellite as it orbits the earth. The thermal control needs of the satellite are found by using this program. This paper will describe the analysis of the satellite performed and results found to date.

THEORY:

The spacecraft thermal balance between cold space (4K) and solar, planetary, and equipment heat sources is the means by which the desired range of equipment and

structural temperatures are obtained. In this analysis, the thermal balance was achieved in the form of a FORTRAN program, that determined the satellite temperature as a function of time.

Before the formulation of the thermal balance program, the important characteristics of the space environment in conjunction with the mathematical techniques to solve the thermal balance had to be considered. In the following discussion, the characteristics of the space environment will be limited to those encountered by a satellite in low earth orbit.

The thermal conductivity of the Earth's atmosphere below an altitude of 90 km is a function of temperature change and is independent of variations in pressure. Above 90 km, the molecular mean free path, or the distance a particle may travel before colliding with another, becomes comparable to the distance in which a temperature change significantly varies, and therefore thermal conductivity becomes pressure dependent. At an altitude of 300 km, the molecular mean free path increases to where thermal conductivity is negligible. Therefore, convective heat transfer may be neglected above 300 km. (Fortescue, 1991)

Hence, heat transfer between the satellite and the space environment above 300 km is limited to radiation heat transfer only. Sources of radiation that may effect the physical temperature of the spacecraft are direct solar radiation, solar radiation reflected from nearby planets(albedo radiation), thermal energy radiated from nearby planets(infrared radiation), and radiation out to deep space from the spacecraft itself.(Fischer, 1993)

Once the sources of radiation have been determined, the radiation needed to be converted into heat transfer. When converting the radiation incident upon the satellite into heat transfer, several parameters must be understood. These parameters consist of the radiation intensity, degree of collimation, and the emissivity and absorptance of the satellite skin. It should be noted that these parameters vary for each type of radiation and are therefore discussed individually.

Solar radiation shall be considered first because it has the highest intensity. This solar radiation intensity, S_0 , from a point source, for any distance D from the sun may be

expressed as shown in Equation 1, where the total power output of the Sun, P , is 3.8×10^{25} W.

$$S_o = \frac{P}{4\pi D^2} \quad (1)$$

The radiation from the Sun may be considered to be collimated, or in other words, entirely made up of parallel waves. This collimation may be assumed due to the radiation originating from a very large distance ($D = 1.50 \times 10^{11}$ m) from the earth. The resulting value of Solar radiation is 1430 W/m^2 for the closest approach to the Sun, which represents the worst case scenario. (Fischer 1993)

In determining the solar radiation heat transfer, Q_{solar} , the product of the solar intensity, area the radiation is incident upon, A_{solar} , and absorptance, α , as shown in Equation 2, must be found.

$$Q_{\text{solar}} = \alpha A_{\text{solar}} S_o \quad (2)$$

Another type of external heat transfer that must be considered is albedo radiation heat transfer. The albedo radiation intensity may be calculated as a fraction of solar radiation, this fraction varies with earth surface conditions. This fraction may be determined by a weighting factor that varies 0.8 to 0.03 for a cloud shrouded or vegetation covered earth, respectively. (Fischer, 1993)

Another consideration in determining the intensity of this type of radiation is the noncollimation caused by the reflection properties of the earth. Non collimated radiation may be taken into account by the use of a view factor. A discussion of the use of a view factor is imperative in determining the albedo radiation incident upon the satellite and therefore follows.

A view factor represents the variation in radiation incident on a surface due to noncollimation of radiation. In order to illustrate the use of a view factor, there are two limiting cases that need to be considered. For the first case, where the satellite would receive the most radiation, the surface of the earth may be thought of as a flat plate extending infinitely in all directions. This infinite flat plate would reflect a fraction of the

solar radiation incident upon it, which has been previously discussed by the use of a weighting factor. The reflected radiation would be non collimated or more specifically reflected in all directions. With the satellite above this infinite flat plate, the radiation incident on a surface of the satellite parallel to the plate, would not only come from the area directly below the satellite, but also from areas far outside. In other words, the radiation from the flat plate is emitted in a disperse pattern, that causes an increase in the radiation flux of the parallel satellite surface. Furthermore, radiation incident on the sides of the satellite due to this disperse pattern, would also add to the radiation flux.

For the second limiting case, the radiation emitted from the infinite flat plate would be collimated, therefore only the parallel surface of the satellite, facing the infinite plate, receives radiation. This second limiting case represents the least amount of radiation incident upon the satellite. The actual case is between these two extremes, the reflected radiation is non collimated and the surface of the earth is neither flat or infinite. By using the view factor, this variation caused by non collimated radiation may be handled.

The intensity of radiation, J_a , received by the satellite therefore depends on the product of view factor, F , solar radiation intensity and weighting factor, a , as shown by Equation 3.

$$J_a = S_0 a F \quad (3)$$

At last, the albedo radiation heat transfer, Q_{albedo} , may be found by the product of the albedo radiation intensity, J_a , the area of the satellite which has radiation incident upon it, A_{albedo} , and the absorptance of the satellite skin, as shown in Equation 4.

$$Q_{\text{albedo}} = \alpha A_{\text{albedo}} J_a \quad (4)$$

The third external heat transfer due to infrared radiation that must be considered, is emitted from the earth acting as a black body radiator. Intensity of the infrared radiation, J_p , is determined by the Stefan-Boltzmann fourth power law, which relates the product of temperature, T , to the fourth power, and the Stefan-Boltzmann constant, σ , with the radiation intensity, as shown by Equation 5.

$$J_p = \sigma T^4 \quad (5)$$

Therefore, to determine the infrared heat transfer, the product of the infrared radiation intensity, the area of the satellite which has radiation incident upon it and the emittance of the satellite skin, ϵ , as shown in Equation 6, must be found.

$$Q_{\text{infra}} = \epsilon A_{\text{infra}} J_p \quad (6)$$

The final heat transfer with the external space environment is from the satellite out to deep space. This heat transfer occurs because the satellite, acting as a black body radiator, emits radiation out to deep space. As previously discussed, a black body radiator's radiation intensity may be found by the Stefan-Boltzmann fourth power law. In determining the heat transfer to space, Q_{internal} , the product of the intensity of the black body radiation, the total emitting area of the satellite, $A_{\text{satellite}}$, the Stefan-Boltzmann constant, and the emittance of the satellite skin, as shown in Equation 7, must be found.

$$Q_{\text{external}} = \epsilon A_{\text{satellite}} \sigma (T_{\text{satellite}}^4 - T_{\text{space}}^4) \quad (7)$$

Once the important characteristics of the space environment have been investigated, a thermal balance of the satellite is determined by setting the heat transfer to deep space from the spacecraft, equal to the solar, albedo and infrared heat transfer. The resulting polynomial is shown in Equation 8,

$$\begin{aligned} \epsilon A_{\text{radiant}} \sigma (T_{\text{satellite}}^4 - T_{\text{space}}^4) = & \alpha A_{\text{solar}} S_o + \alpha A_{\text{albedo}} S_{\text{albedo}} \\ & + \epsilon A_{\text{infrared}} S_{\text{infrared}} \end{aligned} \quad (8)$$

where the term on the left hand side is the heat transfer to deep space, and the terms on the right hand side are the solar, albedo, and infrared heat transfer, listed first, second and third, respectively. The results acquired by this thermal balance may be considered a steady state solution to the energy balance.

The radiations from the different sources, incident on the satellite, vary with the satellites position as it orbits the earth and also as the earth orbits the sun. Hence, a transient analysis was applied to account for the effects of the changing incident radiation on the satellite, with respect to time. This transient analysis was performed by equating the product of the satellite bus mass, the specific heat of the bus, and the change in temperature with respect to time to the sum radiation heat transfers and internal heat generation, Q_{internal} , shown in Equation 9. It should be noted, that the internal heat generation will initially be assumed to be constant.

$$mcp \frac{dT}{dt} = \alpha A_{\text{solar}} S_o + \alpha A_{\text{albedo}} S_{\text{albedo}} + \epsilon A_{\text{infrared}} S_{\text{infrared}} + Q_{\text{internal}} - \epsilon A_{\text{radiant}} \sigma (T_{\text{satellite}}^4 - T_{\text{space}}^4) \quad (9)$$

The resulting equation is a first order nonlinear differential equation, whose solution would be very intensive if solved by hand. Using a numerical solution, such as the Fourth Order Runge-Kutta method, the solution may be simplified. With utilization of a computer program the solution may be found for many different times for many different satellite characteristics, in much less time than one hand calculations.

In order to employ the Runge-Kutta method, Equation 10 had to be transformed into the form

$$T' = F(T,t) \quad (10)$$

where T' is the time derivative of temperature and F is a function of time and temperature. The general solution of this method takes the form shown in Equation 11.

$$y_{i+1} = y_i + \frac{\sum W_j K_j}{\sum W_j} \quad (11)$$

As may be seen from the general form the value of the variable y is calculated for each time increment with the previous y value plus the summation of the weighted gradients, $W_j K_j$, divided by the summation of the weighting factors, W_j . Due to the need for a previous

value y , an initial condition must be specified. It should be noted that as j increases, the order of the Runge-Kutta method increases, increasing the accuracy but also increasing the solution time. For this analysis, the fourth order Runge-Kutta method, which is the most commonly used order, providing adequate accuracy and efficiency, was used.

Now that an external analysis has been completed, an internal analysis may be performed. To begin this analysis, the internal components and equipment may be modeled as nodes with individual temperatures. The heat transfer between these nodes is related by a thermal resistance. To illustrate the relation between two components, Figure 2 has been included. In this figure, R_{12} is the thermal resistance between component 1 and 2

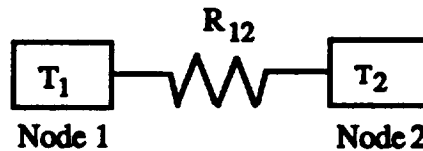


Figure 2 : Thermal Relation Between Two Internal Components

represented by node 1 and 2, respectively. To determine the internal heat transfer between these two component, the quotient of temperature differential an thermal resistance, as shown in Equation 12, must be found.

$$Q_{12} = \frac{T_1 - T_2}{R_{12}} \quad (12)$$

Now that the fundamental concepts of heat transfer between the external environment and satellite, and heat transfer between the internal components have been determined. These external and internal analysis may be combined into one analysis, shown by Figure 3, that takes into account all heat transfers. By relating the external heat transfers from space to the satellite skin and the satellite skin to the internal components of the satellite, a combined analysis may be performed to determine the temperatures of internal components.

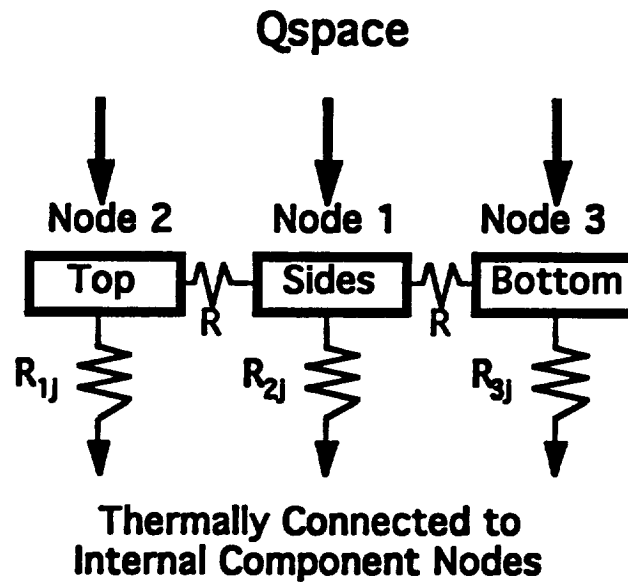


Figure 3: Combined External and Internal Analysis

Computer Program

The analysis outlined in the previous section, leaves the problem of determining a method to solve for the temperature of each node on a satellite over a range of time. It was decided that this problem would be handled easiest by using a computer program to solve for the temperatures on the satellite.

The nodal analysis as outlined in the previous figure, shows, that the space environment affects only the nodes that are representative of the skin of the satellite. The satellite skin is represented by three nodes, one for the top, bottom, and sides of the satellite. The temperatures that the external nodes exhibit, in turn, will be transmitted to the internal components of the satellite depending on the thermal contact between the nodes. Also, the internal nodes can affect the temperature of the outer shell nodes of the satellite through the internal nodes respective heat generation terms. Due to these many interactions between different factors which determine the temperature of each node of the satellite, it seems that a computer program could get rather complex. However, since the satellite is

broken up into nodes, arrays can be used to describe the properties of each of these nodes, and how they interact with the space environment and each other. The array style of the problem leads to a computer program which only needs to solve one function (the temperature function). This temperature function can now be represented in an array format as shown in Equation 13.

$$F = \frac{dT(i)}{dt} = \frac{1}{m(i)C_p(i)} (Q_{int}(i) + Q_{sol}(i) + Q_{alb}(i) + Q_{inf}(i) + \sum_i \sum_j \frac{T(i) - T(j)}{R(i,j)} - \epsilon(i)A(i)\sigma(T(i)^4 - T_o^4)) \quad (13)$$

In this equation, "i" represents the node number being evaluated. The other variables represent either properties of the satellite that would have to be input to the program, or values for heat flux terms that the program must find for each node. To understand this better, each term in the equation will be described. The "T" and "t" represent temperature and time respectively. These are the values that the program will try to find for each node. The mass (m), specific heat (Cp), and internal heat generation (Qint) are values that describe properties of nodes, and must be input to the program. The next terms (Qsol, Qalb, Qinf) represent the different types of radiation from the space environment that will be incident on the external skin of the satellite. For this reason, these variables are set to zero for internal nodes. Also, since the values of these radiations vary with the particular time position of the satellite in orbit, the values will be solved for by the program dependent on the time. The next term describes the heat flux due to the thermal contact between nodes. This term is very important because it is a function of the thermal resistance between nodes, which represents the thermal control system of the satellite. It should be noted that this term has to be calculated for the interaction between every node on the satellite and the node being evaluated. The last term describes the Stefan-Boltzmann

fourth power law for radiation emission from the satellite skin. This term is set to zero for any of the internal nodes.

The temperature function equation and the information outlined in the theory section of this paper were used to formulate a computer program that would solve for the temperature of each node on the satellite as a function of the time in it's orbit. The name of the program is "isat.ftn," and for a detailed description on the operation and background of the program see Appendix B. This appendix contains a program handbook for "isat.ftn." The general form of the program is a Runge-Kutta numerical differential equation solver which was modified to handle arrays. The differential equation solver evaluates the temperature of the satellite in second time steps. At each time step, every node is evaluated by calling different subroutines to find the value of the terms in the temperature function equation. The temperatures of each node of the satellite are then found, and the time step is incremented. The program was initially formulated to analyze the ISAT-1 satellite, but has been modified to accept any satellite configuration that falls under the assumptions used in generating the program (See the Program Handbook). Also, the program could be modified in any way needed to analyze any satellite configuration.

The program that was generated to evaluate the temperature function for each node, needs certain input parameters. These input parameters are entered in the form of an input file which contains orbit information, space environment information, and the thermal properties of the satellite. An example of the input file format can be found in the Program Handbook in the Appendix of this paper.

The orbit information needed is information about the period of the orbit, and the amount of time the satellite will be eclipsed by the earth. The space environment information needed describes the intensity or amount of the different types of radiation that will be incident upon the satellite. The first radiation is the solar radiation which typically is constant in space near the earth. The second type of radiation is the albedo radiation, or the radiation from the sun that is reflected back off of the earth. The input for this radiation is

represented by a decimal fraction of the solar radiation. The third type of radiation is the earth shine, or the infrared radiation that the earth emits due to the fact that it is a thermal emitter and follows the Stefan-Boltzmann fourth power law.

The first satellite properties that must be input are the length (or height) and effective diameter of the satellite. Next, the initial time to begin the analysis and the initial temperature of the satellite must be input. For ease, these input values are usually just assumed to be an initial time of zero, and an initial temperature of 273 K. These values are needed due to the fact that the program uses an ordinary differential solver and must have initial conditions to begin the analysis. The next satellite properties for the input file deal with the nodal analysis of the satellite. First, the number of nodes to be simulated must be input. By allowing the program user to input the number of nodes, many different satellite configurations can be handled by the program, and as a thermal control analysis becomes more developed, the program can handle this expansion. Then, the mass, specific heat, and internal heat generation of each node must be input. The next values describe the properties of the skin or satellite covering. The values needed are the absorptivity (α) and emissivity (ϵ) of the different sides of the satellite (top, bottom, and sides). Also, two view factor values are input. These view factors describe the view factor between the sides of the satellite to earth, and the view factor between the bottom of the satellite to the earth. These view factors are needed in determining the amount of albedo and earth shine radiation incident upon the satellite.

The last properties needed to describe the satellite in the input file, are the thermal resistance, or R-values between the components and between the components and sides of the satellite. The units for the R-values is in Kelvin's per watt. An R-value is needed for each interaction between each node. This means that if two components are not near each other, or are thermally isolated from each other, a large R-value such as 10^9 can be input. Conversely, if two components are in perfect thermal contact, such as between the sides and top and bottom of the satellite, the R-value should be very small. It is suggested that

some kind of preliminary calculation for these R-values be performed because the program will not handle extremely small R-values or values less than 10^{-2} . The values in this situation can be calculated by equation 14.(Chapman, 1984)

$$R = \frac{\Delta x}{k A} \quad (14)$$

where: Δx = conduction distance, m

k = thermal conductivity of the material, W/m-K

A = the area of conduction (or touching), m^2

Due to the set up of the program, an R-value is also needed for the thermal contact between a node and itself, these R-values can just be set to one in the input file.

As was mentioned, these R-values are important because they represent the thermal control system of the satellite. It should be noted that these R-values represent only a passive thermal control system (one without moving parts or a working fluid). For this reason, this program will only evaluate satellite temperatures for passive thermal control systems. The program, however, can be used in two different ways. The first way the program can be used is to determine if a previously designed thermal control system for a satellite is correct. For this type of analysis, the R-values of the materials for the thermal control system can be input into the program, and the high, low, and mean temperature of each node of the satellite will be output. These temperatures can then be compared to the temperature tolerances set for each node (or piece of equipment) to see if the thermal control design is correct. The second way the program can be used is to design a thermal control system. This is performed by assuming R-values for the satellite, running the program, and then modifying the R-values until the temperatures of each node fall within their respective temperature tolerances. If there is not a combination of R-values that satisfies the thermal criteria of the satellite under this situation, this shows that there may be a need for some kind of active thermal control system.

The program was tested by performing an analysis of the ISAT-1 Satellite. At the time the program was developed, not enough specific information about all the satellite and components were known to perform an in depth analysis. However, the main configuration of the satellite was known, and by assuming the properties for the configuration of the satellite, the program was run and tested to see if it would produce results.

RESULTS AND DISCUSSION

For the hypothetical configuration (Fig. 4), the thermal resistance matrix (6×6) contains (36) elements. Symmetry considerations (i.e., $R_{ij} = R_{ji}$) reduce this number to (21). Further reduction of values is achieved by the geometry and known mechanical behavior of the system. For example, the R-value between the top and the bottom of the structure is defined as infinity due to their thermal isolation from one another. Also, the R-value between the top and the sides is defined as ≈ 0 due to their identical material composition and direct contact which results in complete thermal communication. Similar definitions exist which reduce the number of final R-values to be designed for to only (7).

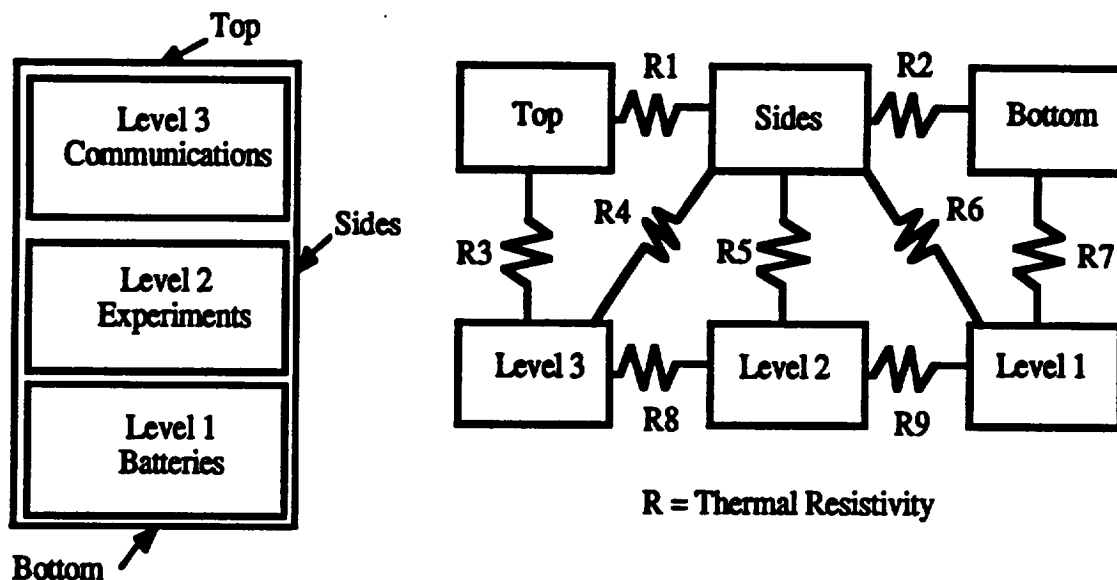


Figure 4: Nodal Configuration

In order to accommodate the actual, potential temperature responses of the nodes during orbit, two worst-case scenarios were investigated. The first of which was to simulate a situation in which all the nodes were in complete thermal isolation. This was achieved by setting the design R-values to ∞ . The temperature responses for this configuration after a twelve-orbit simulation are displayed in Table 3. The design R-values are also indicated.

**Table 3: Node temperatures after 12 orbits -
Complete thermal isolation**

Satellite Component	High Temp. (K)	Low Temp (K)	Mean Temp. (K)
Sides	304.8	280.7	292.7
Top	301.0	279.7	290.3
Bottom	302.0	280.4	291.4
Level 1	360.9	277.5	349.2
Level 2	342.2	323.7	332.9
Level 3	289.1	284.8	286.9

$$R_{14}=R_{15}=R_{16}=R_{26}=R_{34}=R_{45}=R_{56} \sim \infty$$

There are several conclusions which can be drawn from this result. The first is that when the nodes are in thermal isolation, no heat dissipation is allowed; therefore, as the number of orbits increases, the mean temperatures will all steadily increase. This has the potential to cause thermal overloads in the nodes (depending on their individual temperature tolerances). Although very little detailed information is available on component temperature tolerances, information about the batteries is known. Specifically, batteries need to be maintained at around 273 K. The batteries are contained within level 3 which indicates a mean temperature of 349.2 K. Therefore, a second conclusion to be drawn is that the batteries are experiencing thermal overload.

The results after 12 orbits for the opposite worst-case are displayed in Table 4. This case simulates a situation in which the nodes are in complete thermal communication with one another. This scenario was achieved by setting the R-values to ∞ .

**Table 4: Node temperatures after 12 orbits -
Complete thermal communication**

Satellite Component	High Temp. (K)	Low Temp (K)	Mean Temp. (K)
Sides	307.2	301.4	304.3
Top	306.9	301.4	304.1
Bottom	307.0	301.5	304.2
Level 1	307.1	301.7	304.4
Level 2	307.1	301.7	304.4
Level 3	307.0	301.6	304.3

$$R_{14}=R_{15}=R_{16}=R_{26}=R_{34}=R_{45}=R_{56} = \infty$$

The main conclusion here is that when the nodes are allowed to fully dissipate heat to their surroundings (and surrounding nodes) the satellite will eventually reach thermal equilibrium. The potential problem with respect to component temperature tolerances is that temperature differentiation among nodes is prevented which can also result in potential thermal overloads. This phenomenon is evidenced by the fact that the batteries are still in a state of thermal overload at 304.4 K.

Obviously, the two aforementioned cases are unrealistic. In reality one would expect the proper R-values to lie between 0 and ∞ . Owing to the fact that no detailed information is available to direct the convergence upon the proper R-values, a trial and error method was employed with the goal to drive the temperature of the batteries as close to 273 K as possible. The best result possible was achieved with the R-values displayed in Table 5.

**Table 5: Node temperatures after 12 orbits -
Trial & error best case**

Satellite Component	High Temp. (K)	Low Temp (K)	Mean Temp. (K)
Sides	309.5	295.7	302.6
Top	306.2	298.0	302.1
Bottom	307.3	297.2	302.2
Level 1	307.6	299.7	303.7
Level 2	307.5	304.9	306.2
Level 3	306.4	302.9	304.7

$$R_{14}=0.1$$

$$R_{26}=0.1$$

$$R_{56}=0.1$$

$$R_{15}=1E+6$$

$$R_{34}=0.1$$

$$R_{16}=1E+6$$

$$R_{45}=1E+6$$

As indicated, the lowest temperature possible, with this configuration, was 303.7. Since this is still in violation of the tolerance for the batteries, it is concluded that a TCS is necessary for ISAT-1. Manipulation of component location and materials / devices used to affect the heat transfer among the components will be necessary in order to direct the convergence upon the proper R-values necessary to maintain proper temperature requirements. This is the next extension of the TCS design.

REFERENCES

- Chapman, A.J., Heat Transfer. New York: Macmillan Publishing Co., 1984.
- Fisher, W.C., Presentation: Introduction to Satellite Thermal Control, February 18, 1993.
- Fortescue, P and Stark, J. Spacecraft Systems Engineering. West Sussex, England: Wiley, 1991.
- Griffiths, D.V. and Smith, I.M.. Numerical Methods for Engineers. Boca Raton: CRC, 1991.
- Iowa State University of Science and Technology. Summer 1992 Report: Iowa Satellite Project, ISAT-1. The Iowa Satellite Company, Inc., 1992.

APPENDICES

I

Appendix A

Iowa Satellite Project ISAT-1

Iowa State University of Science and Technology

**Summer 1992
Report**

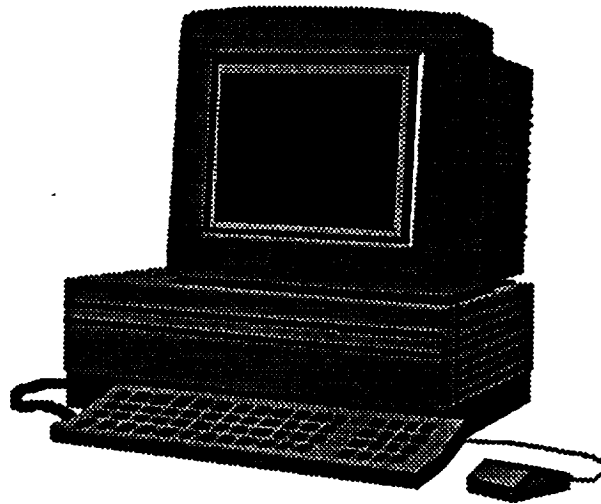
August 20, 1992

Omitted:

Iowa State already possess a copy of this report

Appendix B

Program Handbook



isat.ftn Satellite Thermal Control

Created by:

**Joe Clay
David Prall
Matthew Schneider
Mike Skrbich**

Spring 1993

Thermal Control of ISAT-1

isat.ftn Program Manual

Contents: General operation of thermal control program
and the assumptions made during the synthesis
of the program.

Table of Contents

1.0 Introduction

1.1 Program Background

1.2 Program Method

2.0 Program Operation

3.0 Thermal Control Example

3.1 Example Background

3.2 Input File

3.3 Output

4.0 Program Flow Chart

5.0 Main Program

5.1 Temperature Function

5.2 Ordinary Differential Equation Solver

6.0 Heat Transfer Algorithms

6.1 Solar

6.2 Albedo

6.3 Infrared Radiation

6.4 Thermal Contact Flux

7.0 References

8.0 Appendix: Program Listing

1.0 Introduction

1.1 Program Background

isat.ftn is a fortran program which simulates the temperature fluctuations a satellite will be subjected to as it orbits the earth in space. The program is set up so that the properties of the satellite, as well as orbital information, are input, and the temperature response of different parts of the satellite is output. The program can be used in two general ways. First, the program can be used to determine the thermal control needs of a satellite. Or, secondly, it can be used to determine if the designed thermal control system of a satellite will keep the components of the satellite in their respective temperature tolerances.

As a satellite orbits the earth, it will pass in and out of the shadow of the earth. This causes the different types of radiation incident upon the satellite to vary with time. Since radiation is the only method of heat transfer in space, these fluctuations in the incident radiation will cause the temperatures seen by the skin, or covering, of the satellite to fluctuate. The skin temperature fluctuations will be translated into fluctuations in temperature of the internal components of the satellite. Thermal control of a satellite is needed in order to keep all of the different components of a satellite within temperature tolerances for the safe operation of the equipment. Since, a satellite is generally comprised of many different components, many different temperature needs may exist on one satellite. This brings about the need for an easy method to analyze the temperature of the components of a satellite as it orbits the earth. ISAT.FTN was originally developed to determine the thermal control needs of the ISAT-1 Satellite, but, this program is adaptable for use with other satellites.

In order to use this program for analyzing the temperatures of a satellite, the satellite must be able to fall under some of the assumptions used in generating the program. These assumptions could be modified by making changes to the program, but, without changing the program the satellite must fall within these specifications. First, the satellite must have an orbit height of at least 300 km. This is due to the fact that this program does not take into account the effects of aerodynamic heating, which are existent for orbits below 300 km. Second, it was assumed that the satellite must be able to be modeled as a cylinder with an effective diameter. Most satellites have the general shape of a cylinder (such as hexagonal or other sided geometries) and spin about their center axis. Third, the satellite must orbit with it's bottom side pointing toward the earth. The program was set up this way due to the orbit configuration of the ISAT-1 satellite. This assumption may limit the use of the program for other satellites, but the program can be modified fairly easy by following the outline of the program shown later in this manual. Other assumptions for this program will deal with the input to the program, or the modeling of the satellite properties, which the program needs, to perform calculations. These assumptions will become apparent in the next sections of the manual which describe the program use and show an example of the program capabilities.

Once an input file is prepared for the ISAT.FTN program, the program will run, and two different options for output are available. The first option, is a display of the high, low, and mean temperatures in degrees kelvin, that each satellite component will see during its orbit. The second option is a data output of the temperature response (kelvin) with time of any component on the satellite. This data can be used to make a plot of the temperature response of different components on the satellite during orbit to study carefully the pattern of temperature change of particular components.

1.2 Program Method

The program is set up to perform a nodal heat transfer analysis of the satellite. Each component of the satellite represents a node. This means that there will always be at least three nodes in the analysis because the sides, top, and bottom of the satellite each count as one node. The program simulates the temperature interaction of the space environment with these three nodes and then uses a finite difference method to determine the interaction of the outer skin of the satellite with the internal components, and the internal components with each other. The thermal control system, in this program, is represented by thermal resistance values (R-values) between each internal node and between internal nodes and the external nodes or skin. These R-values are representative of the material properties of either an insulation or some kind of conductor which is to be used in the thermal control system. In this sense, this program can only be used to simulate a passive thermal control system, or one in which there are no mechanical or flowing parts. It can however, point out problem areas that may need active thermal control to maintain the correct temperatures.

As was mentioned, the program can be used in two ways. The first method is to determine the thermal control needs of a satellite. This can be accomplished inputting certain R-values into the program, running it, and then checking the output to determine if the input R-values are sufficient to keep each node within its temperature tolerance. If the R-values are correct, then materials can be chosen to use on the satellite that satisfy the particular R-values. The second method is used to check to see if a thermal control design is correct. In this method, the properties of the satellite and the R-values that are characteristic of the thermal control system design can be input into the program. The temperature output will then show whether or not the R-values (or thermal control design) will keep the satellite components at their respective temperatures.

The next section describes how to run the program, the input that is needed and the input format, and the units of all of the input characteristics.

2.0 Program Operation

The program operation begins by compiling the program. This step is completed by typing "f77 isat.ftn" and pressing return (notice isat.ftn is in lower case letters). Now, the compiler has produced a file called "a.out", and by typing "a.out" and pressing return the program will run. But, before the program can be ran, an input file has to be prepared which contains orbit information, space environment information, and the thermal properties of the satellite.

The orbit information needed is information about the period of the orbit, and the amount of time the satellite will be eclipsed by the earth. The units for the input time is in minutes and the eclipse time input is represented by entering the times that the satellite will enter the shadow and exit the shadow of the earth. The space environment information needed describes the intensity or amount of the different types of radiation that will be incident upon the satellite. The first radiation is the solar radiation which typically is constant in space near the earth. The second type of radiation is the albedo radiation, or the radiation from the sun that is reflected back off of the earth. The input for this radiation is represented by a decimal fraction of the solar radiation. The third type of radiation is the earth shine, or the infrared radiation that the earth emits due to the fact that is a thermal emitter and follows the Stefan-Boltzmann fourth power law. The units for the solar and earth shine radiations must both be in watts per meter squared.

The first satellite properties that must be input are the length (or height) and effective diameter of the satellite. The units on these properties must be in meters. Next, the initial time to begin the analysis and the initial temperature of the satellite must be input. For ease, these input values are usually just assumed to be an initial time of zero, and an initial temperature of 273 Kelvin. These values are needed due to the fact that the program uses an ordinary differential solver and must have initial conditions to begin the analysis. The next satellite properties for the input file deal with the nodal analysis of the satellite. First, the number of nodes to be simulated must be input. Then, the mass, specific heat, and internal heat generation of each node must be input. The units on these values are kilograms, joules per kilogram degree kelvin, and watts, respectively. The next values describe the properties of the skin or satellite covering. The values needed are the absorptivity (alpha) and emissivity (epsilon) of the different sides of the satellite (top, bottom, and sides). Also, two view factor values are input. These view factors describe the view factor between the sides of the satellite to earth, and the view factor between the bottom of the satellite to the earth. These view factors are needed in determining the amount of albedo and earth shine radiation incident upon the satellite.

The last properties needed to describe the satellite are the thermal resistance, or R-values between the components and between the components and sides of the satellite. The units for the R-values is in kelvins per watt. An R-value is needed for each interaction between each node. This means that if two components are not near each other, or are thermally isolated from each other, a large R-value such as 10^6 should be input. Conversely, if two components are in perfect thermal contact, such as between the sides and top and bottom of the satellite, the R-value should be very small. It is suggested that some kind of preliminary calculation for these R-values be performed because the program will not handle extremely small R-values or values less than 10^{-2} . The values in this situation can be calculated by equation 1.[1]

$$R = \frac{\Delta x}{k A} \quad \text{Eq.1}$$

where: Δx = conduction distance, m
 k = thermal conductivity of the material, W/m-K
 A = the area of conduction (or touching), m^2

Due to the set up of the program, an R-value is also needed for the thermal contact between a node and itself, these R-values can just be set to one in the input file.

The format of the input file is set up so that each number has a field of thirteen characters. The format for the input file is shown on the next two pages for a nodal analysis of six nodes. The numbers that are shown on this page need to be included in the input file, and should not be changed. Also, the input values should be entered as decimals or exponential form.

Input File Format

The input file should start at the top of the page of the file:

Orbit Time (min)	Shadow In (min)	Shadow Out (min)
Solar Radiation (w/m^2)	Albedo Factor	Infrared Radiation (w/m^2)
Diameter (m)	Length (m)	Initial Time = 0

Initial Temperature (K)
 Number of Nodes (Integer value including the different sides of the satellite)
 Mass Sides (Node 1) (kg)
 Mass Top (Node 2)
 Mass Bottom (Node 3)
 Mass Node 4
 Mass Node 5
 Mass Node 6
 Specific Heat (cp) Sides (j/kg K)
 Cp Top
 Cp Bottom
 Cp Node 4
 Cp Node 5
 Cp Node 6
 0 Internal Heat Generation Sides (Qint) (W)
 0 Qint Top
 0 Qint Bottom
 Qint Node 1
 Qint Node 2
 Qint Node 3
 Alpha Sides
 Alpha Top
 Alpha Bottom
 Alpha Node 4 = 0
 Alpha Node 5 = 0
 Alpha Node 6 = 0
 Epsilon Sides
 Epsilon Top
 Epsilon Bottom
 Epsilon Node 4 = 0
 Epsilon Node 5 = 0
 Epsilon Node 6 = 0
 View Factor Sides
 View Factor Top = 0
 View Factor Bottom
 View Factor Node 4 = 0
 View Factor Node 5 = 0
 View Factor Node 6 = 0
 R 11 = 1
 R 12 = small value
 R 13 = small value
 R14
 R15
 R16
 R22 = 1
 R 23 = 10^6

$$R_{24} = 10^6$$

$$R_{25} = 10^6$$

$$R_{26}$$

$$R_{33} = 1$$

$$R_{34}$$

$$R_{35} = 10^6$$

$$R_{36} = 10^6$$

$$R_{44} = 1$$

$$R_{45}$$

$$R_{46} = 10^6$$

$$R_{55} = 1$$

$$R_{56}$$

$$R_{66} = 1$$

Once this input file is prepared the following steps are needed to complete a run of the program:

```
%a.out <return>
Input filename:
filename <return>
```

Then the output of the temperatures of the different sides of the satellite and the nodes of the satellite will be output after the program is finished running.

3.0 Thermal Control Example

3.1 Example Background

This example was first generated with preliminary information on the ISAT-1 satellite. The configuration of the satellite is a hexagonal cylinder with an effective diameter of 0.317 meters and a height of 0.643 meters. For the preliminary analysis of this satellite, it was known that the satellite would be broken up into three main shelves: a battery shelf, an experiments shelf, and a communications equipment shelf. This configuration is shown in Figure 1, and thus leads to a six nodal analysis.

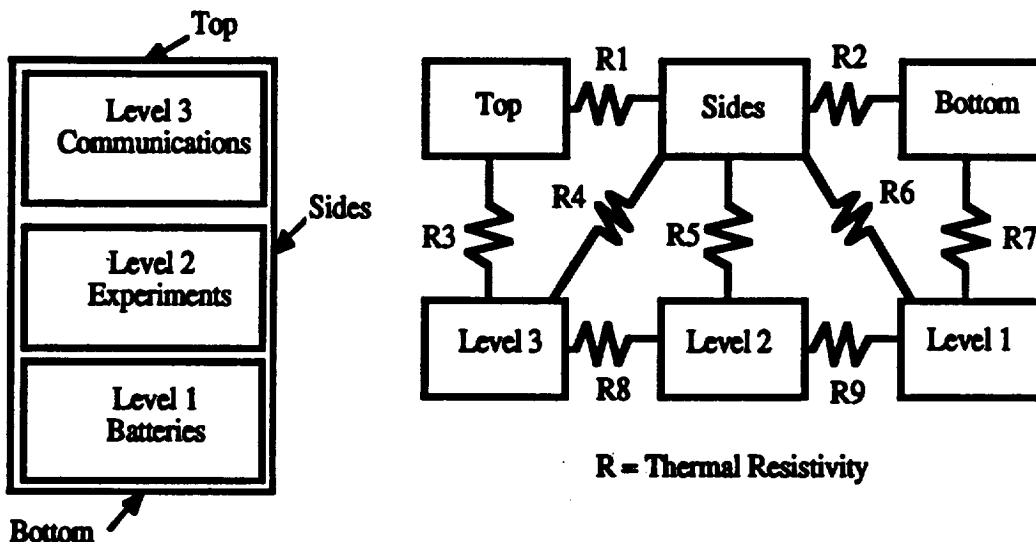


Figure 1: ISAT-1 Preliminary Satellite Configuration

With this configuration, information on each of the six nodes, as outlined in the previous section, was needed in order to run the program. For this particular example, the thermal control system of the satellite was being designed, so, none of the R-values between the nodes were known. Values were assumed for these R-values, and the output for these values will be shown.

3.2 Input File

The orbit for the ISAT-1 satellite was specified as being 95 minutes long with a shadow time of approximately 35 minutes. The height of the orbit was 550 kilometers. These values are reflected in the input file shown on the next two pages.

Information on the different nodes was not completely known at the time of generating this example. However, it was known that the outer shell of the satellite was to be made of 15 kg of Aluminum. The sides of the satellite were to be covered with solar cells, and the top and bottom of the satellite could be covered with whatever is needed for the thermal control of the satellite. For this example, the bottom and top of the satellite were assumed to be covered with white paint ($\alpha = 0.25$, $\epsilon = 0.85$). It should also be noted, that the view factors used in this example were calculated from an average between a view factor for a flat plate above the earth, and a small sphere above the earth.[2] The input file reflects the values that were assumed for the other different properties of the satellite. The nodes for the input file were denoted so that node 4 was the battery shelf, node 5 was the experiments shelf, and node 6 was the communications shelf. This input file was named isat.d, and is shown on the next two pages.

isat.d Input File

95.0	20.0	55.0
1430.0	0.3	230.0
0.317	0.643	0.0
273.0		
6		
10.0		
2.5		
2.5		
25.0		
25.0		
25.0		
900.0		
900.0		
900.0		
700.0		
890.0		
850.0		
0.0		
0.0		
0.0		
18.0		
18.0		
4.0		
0.72		
0.25		
0.25		
0.0		
0.0		
0.0		
0.71		
0.85		
0.85		
0.0		
0.0		
0.0		
0.727		
0.0		
0.727		
0.0		
0.0		
0.0		
1.0		
1.0E-1		
1.0E-1		
1.0E-1		
1.0E9		
1.0E9		
1.0		
1.0E9		
1.0E9		
1.0E9		

1.0E-1
1.0
1.0E-1
1.0E9
1.0E9
1.0
1.0E9
1.0E9
1.0
1.0E-1
1.0

3.3 Output

After the input file was prepared, isat.ftn was compiled and ran. The procedure with output is shown below:

\$ f77 isat.ftn

isat.ftn:

\$ a.out

ENTER INPUT FILENAME:

isat.d

SIDES

THIGH	TLOW	TMEAN
308.44610	295.68741	302.06675

TOP

THIGH	TLOW	TMEAN
304.89307	297.99475	301.44391

BOTTOM

THIGH	TLOW	TMEAN
306.18924	297.21547	301.70235

NODE 4

THIGH	TLOW	TMEAN
306.51317	299.74645	303.12981

NODE 5

THIGH	TLOW	TMEAN
305.81584	304.90375	305.35980

NODE 6

THIGH	TLOW	TMEAN
304.79974	302.94559	303.87267

This output shows the temperatures for each node for the R-values shown in the input file. Notice, that these temperatures are in degrees kelvin. This output shows that with these R-values, there could be some nodes which are at too high of a temperature. By changing the R-values in the input file and running the program, the corresponding change in node temperatures can be found.

Another form of output that can be obtained from the program by adding simple write statements to the program, is the temperature output at each time step for any nodes. This information can be used to make temperature verses time plots for any node of the satellite. Figure 2 shows a graph of this type, for the sides (Node 1) of the satellite described by this example.

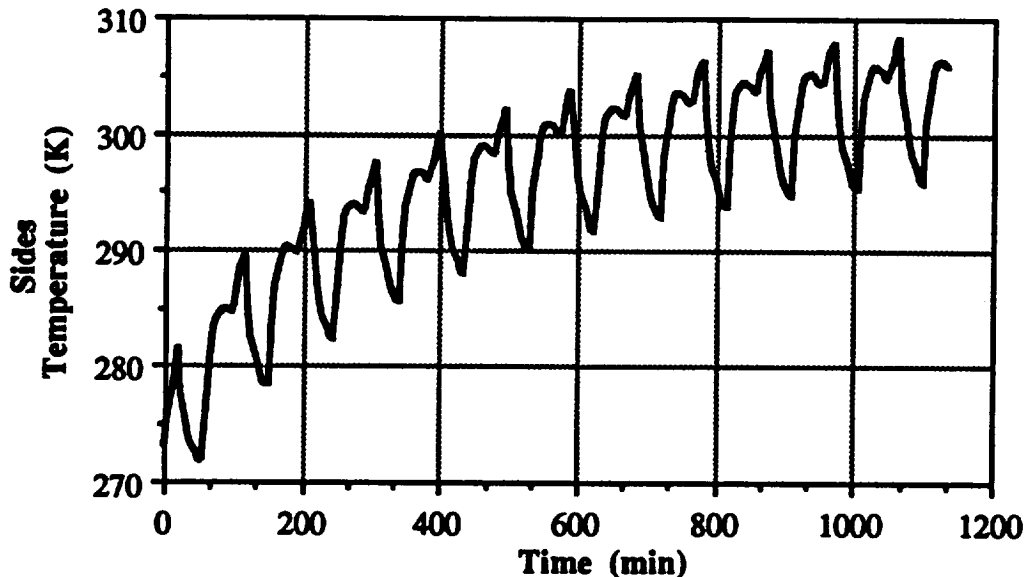


Figure 2: Temperature Response of Sides of Satellite

It can be seen from this graph, that the temperature of the sides starts out at the initial temperature of 273 Kelvin, and then as the satellite continues going through orbits, the temperature reaches an equilibrium fluctuation of temperature. The program is designed to take this equilibrium condition into account before reading final high, low, and mean temperatures for each node of the satellite.

4.0 Program Flow Chart

The program described by this handbook was developed to study the thermal control needs of a satellite. A satellite in space, will interact with the space environment which will cause the temperature of the satellite skin to change. These changes in skin temperature will cause changes in the temperature of different components of the satellite depending on the thermal contact between the components and the skin and the components with each other. The equation that describes the transient thermal response of a body in space is a first order nonlinear differential equation. This program was developed so that the temperature of the satellite would be evaluated at each second time step in its orbit. Due to this fact, a numerical differential equation solver was implemented. The differential equation solver and temperature function will be discussed further in the next section of this manual.

Since a satellite is comprised of many components, the temperature of each component of the satellite has to be evaluated. In order to perform this, the satellite can be broken up into nodes, and the temperature of each node can be evaluated using the differential equation solver. The whole analysis results in a program that functions by using arrays to describe the different properties of each node of the satellite. The problem of finding the temperature at each node is handled by incrementing the node numbers, and solving the temperature function for each node at each time step. Obviously, only the first three nodes (the skin of the satellite) will be affected in different ways by the external space environment, and will radiate heat to space. While, with the other nodes (internal components) the temperature function will remain the same. These different types of heat flux on a node were handled by calling different subroutines to evaluate all of the heat flux

on a node i.e. for internal components, heat flux to and from the space environment was set to zero.

A flow chart of the program is shown in Figure 3. This flow chart shows the general method that the program follows to solve for the temperature of different parts of a satellite at each time step in it's orbit. It should be noted that the program is set to simulate a certain number of orbits in order to reach an equilibrium fluctuation of temperature during an orbit. The output temperatures of the satellite are not evaluated in the program until the equilibrium condition has been reached.

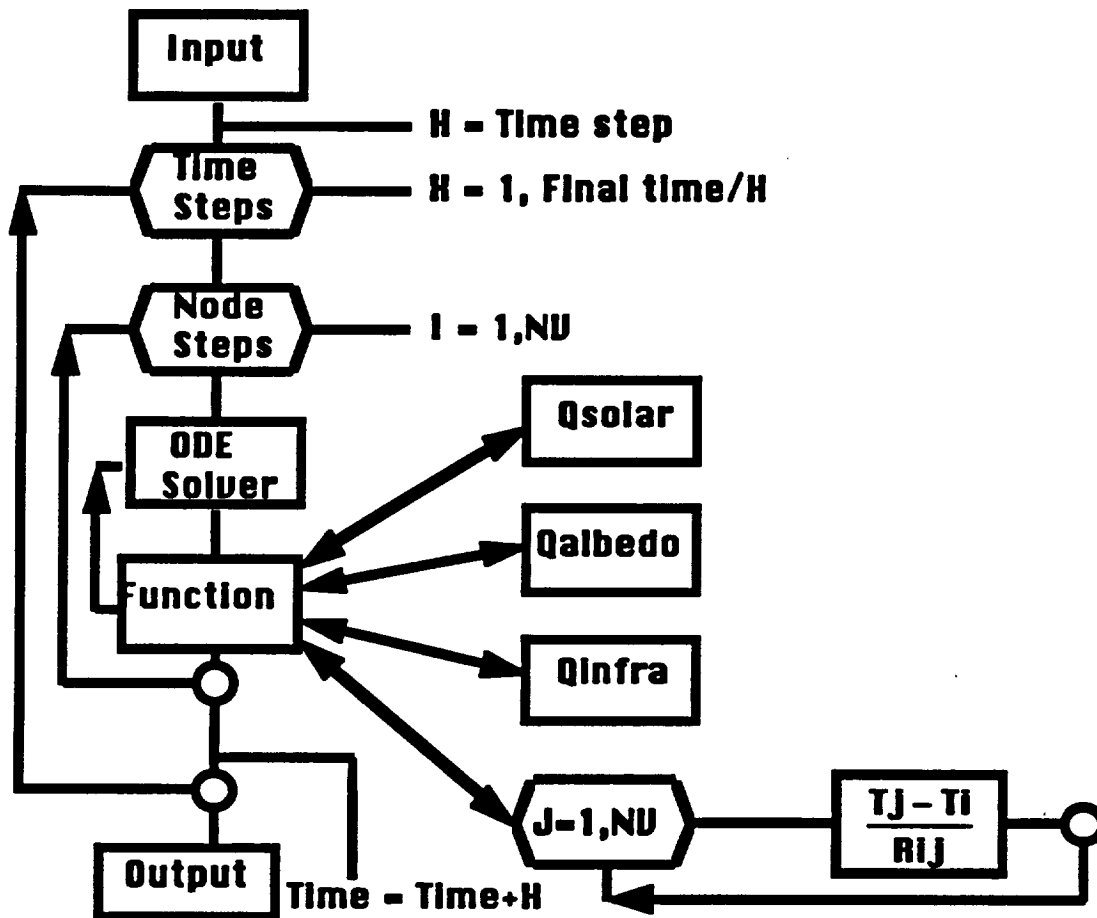


Figure 3: isat.ftn Program Flow Chart

5.0 Main Program

5.1 Temperature Function

The main point of this program is to solve a temperature function for each node of the satellite. This temperature function is a first order nonlinear differential equation that is solved for each node at each time step in the orbit of the satellite. The temperature function is shown below in equation 2.

$$F = \frac{dT(i)}{dt} = \frac{1}{m(i) C_p(i)} (Q_{int}(i) + Q_s(i) + Q_a(i) + Q_{inf}(i) +$$

$$\sum_i \sum_j \frac{(T(j) - T(i))}{R(i,j)} - \epsilon(i) A(i) \sigma (T(i)^4 - T_o^4))$$

Eq. 2

From equation 2, it can be seen that there are many different factors which affect the nodes of the satellite configuration. The first terms are the solar, albedo, and infrared radiation heat fluxes to the external skin of the satellite. The functions for these heat fluxes are described in more detail in the next section of this manual. For all internal nodes, these values are set to zero. Another term, is the internal heat generation of each node. These values are input to the program and are typically zero for the external nodes of the satellite. The next term deals with the heat flux due to thermal contact between nodes. This term takes the temperature difference between each node, divides this by the thermal resistance between the nodes, and determines the heat flux into or out of the node in question. The last term in this equation, is the satellite radiation to space. This radiation is governed by the Stefan-Boltzmann fourth power law, and only is taken into account for the external nodes of the satellite.

It is seen from equation 2, that the temperature function is written in terms of arrays. This allows the program to iterate to evaluate each node at each time step in the solving of the temperature of the satellite. The format of this equation also makes it easy to change the factors that are in the temperature function without changing the function itself, if it is desired to modify the program.

5.2 Ordinary Differential Equation Solver

The ordinary differential equation solver used in this program is the Runge-Kutta numerical differential equation solving scheme.[3] The form of the differential equation solver was modified to adapt to the arrays used in the program. The general form of the Runge-Kutta method is:

$$y' = F(x,y)$$

$$y_{i+1} = y_i + \frac{\sum W_j K_j}{\sum W_j}$$

where: y = Temperature
 x = Time
 W = Gradient factors
 K = Time step

In the program, the differential equation solver is set up to solve for the temperature of each node in terms of second increments in time. This time increment results in a program running time that is very long. Without sacrificing accuracy, a time step of every ten seconds was implemented in the program to decrease the running time. This time step is represented by the variable "H" in the listing of the program found in the Appendix of this manual. It is recommended that this time step not be increased any more than thirty

seconds, because there is a trade off with the accuracy of the program due to the fact that the R-values input in the program cannot be as large or as small as with a small time step. Reference 4 explains this "stability criteria" in more detail.[4]

6.0 Heat Transfer Algorithms

The purpose of this section is to describe the method used to describe the different types of heat transfers that are in the temperature function. The first three parts to this section describe the radiation heat transfer between the space environment and the outside of the satellite. The general equation which governs these heat transfers, or heat fluxes, is shown in equation 3.

$$Q_{rad} = J * A_J * Alpha * View Factor * Area Factor \quad Eq. 3$$

where: J = Radiation intensity (W/m^2)

A_J = Satellite area affected by the radiation (m^2)

$Alpha$ = Absorptivity of the satellite skin

View Factor = 1 for Solar, and other values for albedo and infrared

Area Factor = Factor which describes the amount of projected area of the satellite surface, depending on orientation

From this equation, the intensity and alpha values, and view factors are input into the program, and the areas are calculated from the dimensions input into the program. The last variable, the area factor, is what comprises most of the differences for each type of radiation on the different sides of the satellite. These factors are different for each type of radiation and each section of the outside of the satellite. The factors were developed for the ISAT-1 satellite which has an orbit orientation described by Figure 4. From figure 4, it can be seen that the radiation incident on the different sides of the satellite will vary depending on the time of the orbit and the orientation of the satellite at that time.

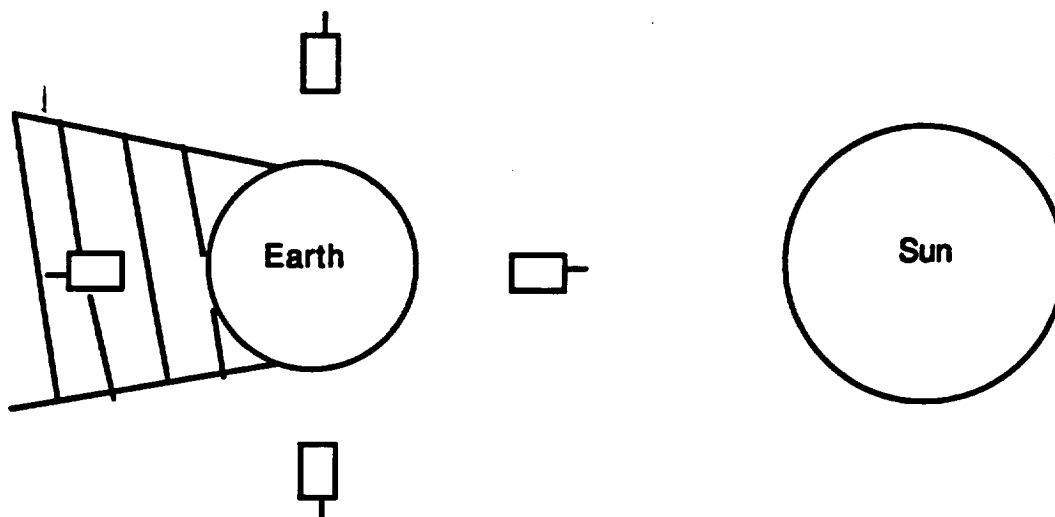


Figure 4: Satellite Orientation During Orbit

The other heat transfer factors in the temperature function are fairly self explanatory, but, the method for handling the heat flux due to thermal contact between two nodes will also be discussed briefly in this section. For more information on the different types of radiation explained in these sections and in the temperature function of equation 3, see Reference 5.[5]

6.1 Solar

The solar radiation will only be incident on the top of the satellite and the sides of the satellite as can be seen in Figure 4. For this reason, the area factor for the bottom of the satellite was set to zero. The top of the satellite will receive 100 percent solar radiation when it is perpendicular to the sun and receive 0 percent of the radiation in the shadow of the earth. In between these two, the incident radiation was assumed to decrease and increase in a linear fashion. This is depicted in Figure 5, which shows the top area factor for solar radiation for the period of one orbit. This particular orbit is 95 minutes long with a 35 minute shadow time, but the program will handle calculating the area factor for other orbit parameters. The equations that govern all of the different area factors used in this program can be found in the program listing in the Appendix of this manual. The different area factor equations are found in their respective radiation subroutines in the listing.

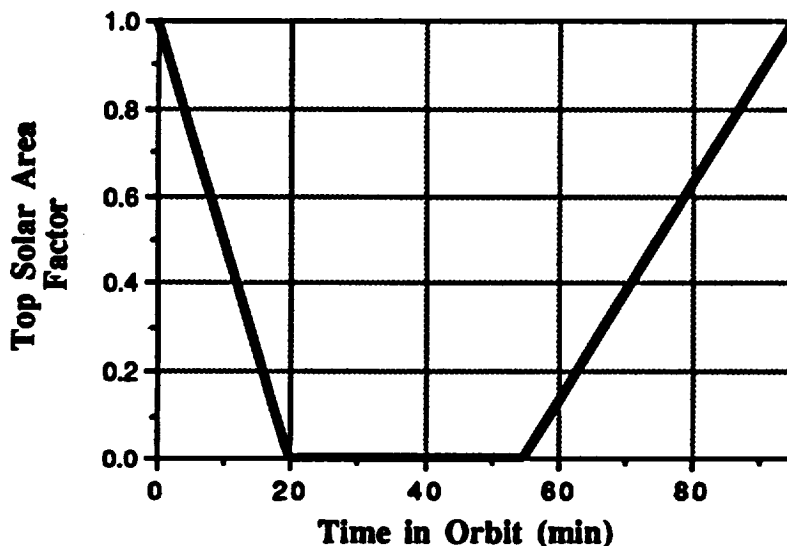


Figure 5: Solar Area Factor for Top of Satellite

The area factor for the side of the satellite was handled in the same way as the top factor, but the sides of the satellite receive 100 percent radiation just before entrance into the shadow of the earth, and receive 0 percent radiation perpendicular to the sun. The side solar area factor is depicted by Figure 6.

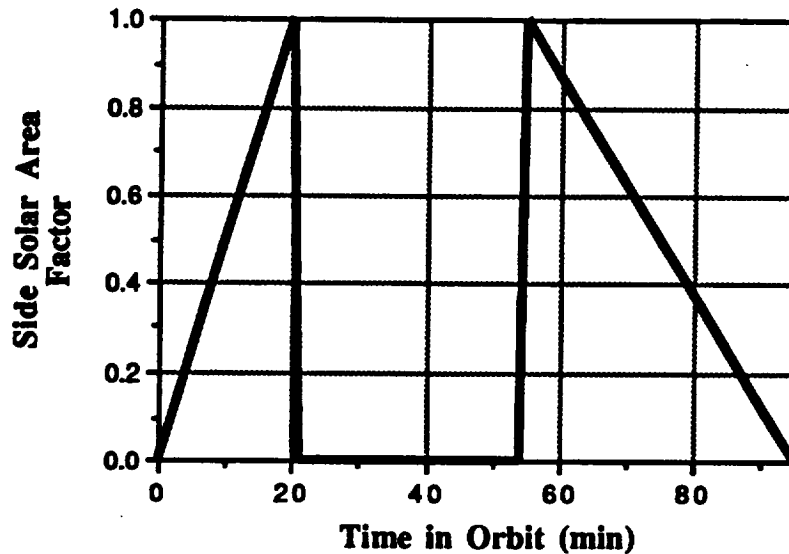


Figure 6: Solar Area Factor for Sides of Satellite

6.2 Albedo

The albedo radiation is solar radiation reflected off of the earth back into space. For this reason, the albedo radiation will only be incident upon the bottom of the satellite and the sides of the satellite. The area factor for albedo for the top of the satellite was thus set to zero. The area factors for the other two sides was handled a little differently than the solar factor. As the satellite comes close to entering the shadow of the earth, the albedo will start to decrease due to the fact that the shadow will not reflect to the satellite. Also, after the satellite has entered the shadow, there will still be a small amount of albedo affecting the satellite. For this reason, the albedo area factor was decreased starting ten minutes before entering the shadow of the earth, and does not reach zero until five minutes after entry into the shadow. This is depicted by Figure 7. It should be mentioned, that in this program, the albedo area factor carried the same value for both the bottom and sides of the satellite.

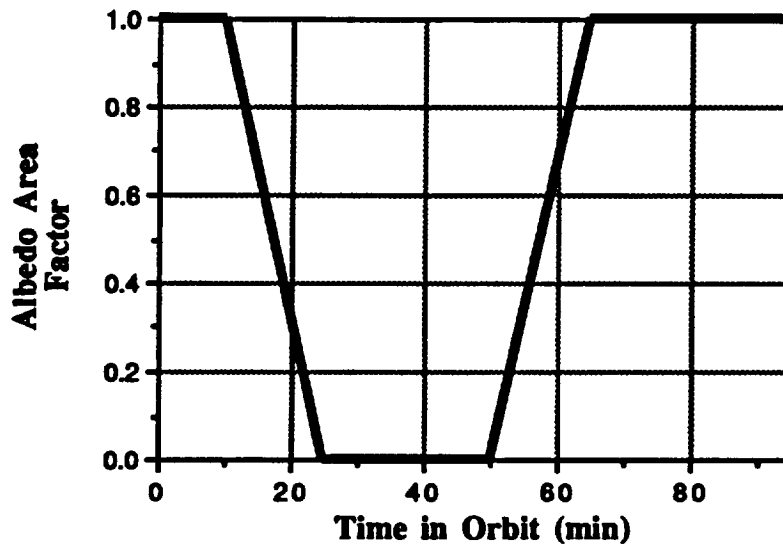


Figure 7: Albedo Area Factor for Sides and Bottom of Satellite

6.3 Infrared Radiation

The infrared radiation incident on a satellite remains constant due to the fact that the earth is always emitting this radiation throughout the entire orbit. The infrared radiation, however, will only affect the bottom and sides of the satellite. Because a view factor is already used for determining the heat flux due to this radiation on the satellite, and because the radiation is a constant throughout the orbit, the infrared radiation area factor was set to one for the bottom and sides of the satellite. This area factor was set to zero for the top of the satellite because it does not receive any of the radiation.

6.4 Thermal Contact Flux

It is important to understand the method used for determining the thermal contact heat flux between nodes for this program, because it depends on the R-values, which are the thermal control system. The general equation for finding the heat transfer between two nodes and different temperatures separated by a thermal resistor is seen in equation 4.

$$Q_{12} = \frac{T_1 - T_2}{R_{12}} \quad \text{Eq. 4}$$

where: T = Temperature of the node (K)
 R = Thermal resistance between nodes (K/W)
 Q = Heat Transfer between nodes (W)

In the program, the heat flux between nodes was evaluated for each node at each time step. This procedure can be seen in the temperature function subroutine of the program listing found in the Appendix of this manual. For more information on the method for performing a nodal analysis as explained by equation 4, see Reference 6.[6]

C-4

7.0 References

- [1] Chapman, Alan J. Heat Transfer. New York: Macmillan Publishing Company, 1984, p. 39.
- [2] Fortescue, Peter and Stark, John. Spacecraft Systems Engineering. West Sussex, England: Wiley, 1991, p. 280-287.
- [3] Griffiths, D.V. and Smith, I.M.. Numerical Methods for Engineers. Boca Raton: CRC, 1991, p. 219-224.
- [4] Chapman, Alan J. Heat Transfer. New York: Macmillan Publishing Company, 1984, p. 160 - 167.
- [5] Clay, Joseph. Prall, David. Schneider, Matthew and Skrbich, Mike. Design of a Thermal Control System for the ISAT-1 Satellite. University of Iowa, 1993.
- [6] Chapman, Alan J. Heat Transfer. New York: Macmillan Publishing Company, 1984, p. 130 - 160.

8.0 Appendix: Program Listing

```

      PROGRAM THERMALCONTROL
C   THERMAL CONTROL DESIGN PROGRAM FOR THE ISAT-1 SATELLITE
C
C   THERMAL CONTROL GROUP
C
C   UNIVERSITY OF IOWA
C
C   GROUP MEMBERS:  JOE CLAY, DAVE PRALL, MATT SCHNEIDER,
C                   MIKE SKRBICH
C
C   SPRING 1993
C

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SM(50), CP(50), Q(50), T(50), ALPHA(50), EPS(50)
      DIMENSION A(50), TH(50), TL(50), TM(50), VF(50), R(50,50)
      DIMENSION TSUB(50)
      COMMON/ORPAR/ORB,TORBCO,SHADEIN,SHADEOUT,TIMEORB
      COMMON/OTHPAR/TCON,APROJ,SIGMA,SO,RINFRA,ALB,TC2
      COMMON/VAR/NV

C   READ INPUT FILE
C
      CHARACTER*8, INNAME
      PRINT *, 'ENTER INPUT FILENAME:'
      READ '(A)', INNAME
      OPEN (UNIT = 1, FILE = INNAME, STATUS = 'OLD')
      READ (1,2) TIMEORB, SHADEIN, SHADEOUT
      READ (1,2) SO, ALB, RINFRA
      READ (1,2) DIA, SLEN, TIME
      READ (1,3) T(1)
      DO 7 I= 2,6
        T(I) = T(1)
7      CONTINUE
900    READ (1,900) NV
      FORMAT (I3)
      DO 5 IO = 1,6
        DO 6 IP = 1,NV
          IF (IO.EQ.1) THEN
            READ (1,3) SM(IP)
          ENDIF
          IF (IO.EQ.2) THEN
            READ (1,3) CP(IP)
          ENDIF
          IF (IO.EQ.3) THEN
            READ (1,3) Q(IP)
          ENDIF
          IF (IO.EQ.4) THEN
            READ (1,3) ALPHA(IP)
          ENDIF
          IF (IO.EQ.5) THEN
            READ (1,3) EPS(IP)
          ENDIF
          IF (IO.EQ.6) THEN

```

```

        READ (1,3) VF(IP)
        ENDIF
6      CONTINUE
5      CONTINUE
      K = 0
      L = 1
4      IF (K.EQ.NV) THEN
        L = L + 1
        K = L - 1
      ENDIF
      K = K + 1
      READ (1,3) R(L,K)
      R(K,L) = R(L,K)
      IF (L.LE.NV-1) THEN
        GO TO 4
      ENDIF
2      FORMAT (3F13.5)
3      FORMAT (F13.5)
      CLOSE (1)
C
C  CONSTANTS
C    TCON CONVERTS TIME TO MINUTES IN THE TEMPERATURE FUNCTION
C    SIGMA IS THE STEFAN BOLTZMAN CONSTANT
C    ORB AND TORBCO ARE USED TO COUNT ORBITS IN THE PROGRAM
C  DELTA TIME = 1 SEC

      TC2 = 60.0D0
      TIMEORB = TIMEORB * TC2
      SHADEIN = SHADEIN * TC2
      SHADEOUT = SHADEOUT * TC2
      ORBITS = 12.0D0
      SIGMA = 5.6703D-8
      ORB = 1.0D0
      TORBCO = TIMEORB
      TCON = 1.0D0

C  MAIN
C    AREAS CALCULATES DIFFERENT AREAS FROM INPUT DIMENSIONS

      CALL AREAS (DIA, SLEN, A, APROJ)

C  ODE SOLVER: RUNGE-KUTTA METHOD

      H = 10
      NSTEPS = (TIMEORB*ORBITS)/H
      KPRINTT = 300
      DO 10 I = 1, NV
        TH(I) = 0.0D0
        TL(I) = 1.0D3
10     CONTINUE
      DO 20 Y = 1, NSTEPS
C       IF (KPRINTT.EQ.300) THEN
C         WRITE (6,101) T(4)
C         KPRINTT = 10
C       ELSE
C         KPRINTT = KPRINTT + 10
C       ENDIF
C       CALL PRLONG (PRINTT, T)

```

```

      IF (ORB.GE.12.0D0) THEN
        DO 30 ICOUN = 1,NV
          IF (T(ICOUN).GT.TH(ICOUN)) THEN
            TH(ICOUN) = T(ICOUN)
          END IF
          IF (T(ICOUN).LT.TL(ICOUN)) THEN
            TL(ICOUN) = T(ICOUN)
          END IF
30      CONTINUE
        END IF
        DO 40 I = 1,NV
          CALL VECADD(T,0,TSUB)
          RK0 = H*F(TIME,TSUB,I,R,Q,EPS,A,CP,SM,ALPHA,VF)
          CALL RES(TSUB)
          CALL VECADD(T,0.5D0*RK0,TSUB)
          RK1 = H*F(TIME+0.5D0*H,TSUB,I,R,Q,EPS,A,CP,SM,ALPHA,VF)
          CALL RES(TSUB)
          CALL VECADD(T,0.5D0*RK1,TSUB)
          RK2 = H*F(TIME+0.5D0*H,TSUB,I,R,Q,EPS,A,CP,SM,ALPHA,VF)
          CALL RES(TSUB)
          CALL VECADD(T,RK2*1.0D0,TSUB)
          RK3 = H*F(TIME+H,TSUB,I,R,Q,EPS,A,CP,SM,ALPHA,VF)
          T(I) = T(I) + (RK0+2.0D0*RK1+2.0D0*RK2+RK3)/6.0D0
40      CONTINUE
        TIME = TIME + H
20      CONTINUE
        DO 50 IZ = 1,NV
          TM(IZ) = (TH(IZ)+TL(IZ))/2.0D0
          WRITE (6,*)
          IF (IZ.EQ.1) WRITE(*,*) '          SIDES'
          IF (IZ.EQ.2) WRITE(*,*) '          TOP'
          IF (IZ.EQ.3) WRITE(*,*) '        BOTTOM'
          IF (IZ.GT.3) WRITE(*,*) '      NODE ',IZ
          WRITE (6,*) ('      THIGH          TLOW          TMEAN')
          WRITE (6,101) TH(IZ),TL(IZ),TM(IZ)
50      CONTINUE
100     FORMAT (3E13.5)
101     FORMAT (3F13.5)
      END

```

```

C
C SUBROUTINES FOR SAT. HEAT TRANSFERS
C
C TEMPERATURE FUNCTION

```

```

      FUNCTION F(TIME,T,I,R,Q,EPS,A,CP,SM,ALPHA,VF)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ORPAR/ORB,TORBCO,SHADEIN,SHADEOUT,TIMEORB
      COMMON/OTHPAR/TCON,APROJ,SIGMA,SO,RINFRA,ALB,TC2
      COMMON/VAR/NV
      DIMENSION R(50,50),Q(50),EPS(50),A(50),SM(50),CP(50),T(50),VF(50),
&ALPHA(50)
C      INTEGER I

      CALL ORBCOUNT(TIME,ORB,TORBCO,TIMEORB,TIME2)
      CALL SOLAR(I,TIMEORB,SO,TIME2,SHADEIN,SHADEOUT,A,ALPHA,
&APROJ,QSOLAR)
      CALL ALBEDO(I,ALB,SO,TIME2,SHADEIN,SHADEOUT,ALPHA,VF,A,TC2,

```

```

      &QALBEDO)
      CALL INFRARED(I,EPS,A,VF,RINFRA,QINFRA)

      TDR = 0.0D0
      DO 60 J = 1,NV
        TDR = TDR + ((T(J)-T(I))/R(I,J))
60      CONTINUE
119     FORMAT(F13.5)

      F = ((Q(I) + QSOLAR+ QALBEDO+ QINFRA+ TDR - SIGMA * EPS(I) *
&A(I) * T(I)**4.0D0) * TCON) / (SM(I)*CP(I))

12     FORMAT(6F13.5)
      RETURN
      END

C LAUNCH

C ORBCOUNT
      SUBROUTINE ORBCOUNT(TIME,ORB,TORBCO,TIMEORB,TIME2)
      IMPLICIT REAL*8(A-H,O-Z)

      IF (TORBCO.LT.TIME) THEN
        ORB = ORB + 1.0D0
        TORBCO = ORB * TIMEORB
      ENDIF

      TIME2 = TIME - TIMEORB*(ORB-1.0D0)

      RETURN
      END

C AREAS
      SUBROUTINE AREAS(DIA,SLEN,A,APROJ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(50)
      COMMON/VAR/NV

      PI = 3.14159D0
      A(1) = PI * DIA * SLEN
      A(2) = PI * ((DIA**2.0D0)/4.0D0)
      A(3) = A(2)
      DO 301 IAO=4,NV
        A(IAO) = 0.0D0
301     CONTINUE
      APROJ = DIA * SLEN

      RETURN
      END

C SOLAR RADIATION
C ONLY EFFECTS THE SHELL TOP AND SIDES I.E. I = 1,2
      SUBROUTINE SOLAR(I,TIMEORB,SO,TIME2,SHADEIN,SHADEOUT,
&A,ALPHA,APROJ,QSOLAR)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(50),ALPHA(50)

      IF (I.GE.3) THEN

```

```

      QSOLAR = 0.0D0
    ENDIF

    IF (I.EQ.1) THEN
      IF (TIME2.LE.SHADEIN) THEN
        FAC = TIME2/SHADEIN
      ENDIF
      IF (TIME2.GE.SHADEOUT) THEN
        FAC = -(TIME2/(TIMEORB-SHADEOUT))+1.0D0+(SHADEOUT/
&      (TIMEORB-SHADEOUT))
      ENDIF
      IF ((TIME2.GT.SHADEIN).AND.(TIME2.LT.SHADEOUT)) THEN
        FAC = 0.0D0
      ENDIF
      QSOLAR = ALPHA(1) * SO * APROJ * FAC
    ENDIF

    IF (I.EQ.2) THEN
      IF (TIME2.LE.SHADEIN) THEN
        FAC = -(TIME2/SHADEIN) + 1.0D0
      ENDIF
      IF (TIME2.GE.SHADEOUT) THEN
        FAC = (TIME2/(TIMEORB-SHADEOUT))+1.0D0-(TIMEORB/
&      (TIMEORB-SHADEOUT))
      ENDIF
      IF ((TIME2.GT.SHADEIN).AND.(TIME2.LT.SHADEOUT)) THEN
        FAC = 0.0D0
      ENDIF
      QSOLAR = ALPHA(2) * SO * A(2) * FAC
    ENDIF

    RETURN
  END

```

C INFRARED RADIATION

```

C ONLY EFFECTS THE SIDES AND BOTTOM I.E. I = 1,3
  SUBROUTINE INFRARED(I, EPS, A, VF, RINFRA, QINFRA)
    IMPLICIT REAL*8 (A-H, O-Z)
    DIMENSION A(50), VF(50), EPS(50)

    QINFRA = EPS(I) * A(I) * VF(I) * RINFRA

    RETURN
  END

```

C ALBEDO RADIATION

```

C ONLY EFFECTS THE SIDES AND BOTTOM I.E. I = 1,3
  SUBROUTINE ALBEDO(I, ALB, SO, TIME2, SHADEIN, SHADEOUT, ALPHA, VF
&  , A, TC2, QALBEDO)
    IMPLICIT REAL*8 (A-H, O-Z)
    DIMENSION A(50), VF(50), ALPHA(50)

    S1IN = SHADEIN - 10.0D0 * TC2
    S2IN = SHADEIN + 5.0D0 * TC2
    S1OUT = SHADEOUT - 5.0D0 * TC2
    S2OUT = SHADEOUT + 10.0D0 * TC2

    FAC2 = 1.0D0

```

```

      IF ((TIME2.GT.S1IN).AND.(TIME2.LT.S2IN)) THEN
        FAC2 = -(TIME2/(15.0D0*TC2)) + 1.0D0 +
&      ((S1IN)/(15.0D0*TC2))
      ENDIF
      IF ((TIME2.GT.S1OUT).AND.(TIME2.LT.S2OUT)) THEN
        FAC2 = (TIME2/(15.0D0*TC2)) + 1.0D0 -
&      ((S2OUT)/(15.0D0*TC2))
      ENDIF
      IF ((TIME2.GE.S2IN).AND.(TIME2.LE.S1OUT)) THEN
        FAC2 = 0.0D0
      ENDIF

```

```

      QALBEDO = SO * ALB * ALPHA(I) * A(I) * VF(I) * FAC2

```

```

      RETURN
      END

```

C

```

      SUBROUTINE VECADD(T, SCAL, TSUB)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION T(50), TSUB(50)
      COMMON/VAR/NV

```

```

      DO 1 KP = 1, NV
1      TSUB(KP) = T(KP) + SCAL

```

```

      RETURN
      END

```

C

```

      SUBROUTINE RES(TSUB)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION TSUB(50)
      COMMON/VAR/NV

```

```

      DO 133 KL = 1, NV
133      TSUB(KL) = 0.0D0

```

```

      RETURN
      END

```